Australian Government
Department of Defence
Science and Technology

Dialogue Systems & Dialogue Management

Deeno Burgan

National Security & ISR Division
Defence Science and Technology Group

DST-Group-TR-3331

ABSTRACT

A spoken dialogue system (SDS) is a specialised form of computer system that operates as an interface between users and the application, using spoken natural language as the primary means of communication. The motivation for spoken interaction with such systems is that it allows for a natural and efficient means of communication. It is for this reason that the use of an SDS has been considered as a means for furthering development of DST Group's Consensus project by providing an engaging spoken interface to high-level information fusion software. This document provides a general overview of the key issues surrounding the development of such interfaces.

RELEASE LIMITATION

Approved for public release

Produced by

National Security & ISR Division
DST Group Edinburgh
PO Box 1500
Edinburgh SA 5111

Telephone: 1300 333 362

January 2017
AR-016-776

APPROVED FOR PUBLIC RELEASE

Dialogue Systems & Dialogue Management

Executive Summary

A spoken dialogue system is a specialised form of computer system that operates as an interface between users and the application, using spoken language as the primary means of communication. This report provides a general overview of some of the key themes identified in the area of spoken dialogue systems and their dialogue management capability. We discuss the strengths and disadvantages of some of the approaches in the presented theory, identifying possibilities for future research. We also provide an evaluation of the suitability of the theory presented as relevant to the informed development of a dialogue management capability within DST Group's Consensus project.

Author

Deeno Burgan

National Security & ISR Division

Deeno Burgan is a computer science undergraduate from Swinburne University of Technology. He joined DST Group Edinburgh in 2016 as part of a twelve month industry experience placement to research spoken dialogue systems. His interests include GNU/Linux systems, cyber security, and any technology that brings the world closer to a cyberpunk dystopia.

Contents

1. INTRODUCTION..1

2. BACKGROUND..2

2.1 Consensus..2
2.2 Requirements..2

3. METHODOLOGY..3

3.1 Survey Process..3
3.2 Selection Criteria..4
3.3 Report Structure..5

4. SPOKEN DIALOGUE SYSTEMS..6

4.1 Definition..6
4.2 Components..6
4.2.1 Automatic Speech Recogniser..6
4.2.2 Natural Language Understanding..7
4.2.3 Dialogue Manager..7
4.2.4 Natural Language Generation..9
4.2.5 Text-to-speech..10
4.3 Multimodal Interaction..10
4.3.1 Embodied Conversational Agents..11
4.3.2 Visual..13
4.3.3 Emotion..13
4.4 History of SDSs..14

5. DIALOGUE MANAGEMENT..17

5.1 Context..17
5.2 Turn-taking..18
5.3 Action Selection..20
5.3.1 Definitions..20
5.3.2 Handcrafted..20
5.3.2.1 Finite-state Machines..21
5.3.2.2 Information State..21
5.3.2.3 Rule-based..22
5.3.2.4 Frame-based..23
5.3.2.5 Plan-based..24
5.3.2.6 Agent-based..25
5.3.3 Machine Learning..26
5.3.3.1 Bayesian Networks..27
5.3.3.2 Neural Networks..28
5.3.3.3 Markovian Models..28
5.3.4 Summary of Strategies..30
5.4 Multi-agent Architectures..31
5.4.1 Agent Intelligence..32
5.4.2 Cooperating and Negotiating Agents..33
5.5 Understanding the User..33
5.5.1 Intent & Goal Recognition..34

5.5.1.1 Definitions 34
5.5.1.2 Application 35
5.5.2 User Modelling 35

5.5.2.1.1 Terminology 35

5.5.2.2 What's in a Model? 36
5.5.2.3 Issues with Modelling 37
5.6 Error Handling 38
5.6.1 Why Handle Errors? 38
5.6.2 Types of Errors 38

6. EVALUATION OF DM TECHNIQUES 40

6.1 Action Selection Methods 40
6.2 Dialogue with Agents 41
6.3 Applicability to Consensus 42
6.3.1 Choosing an Action Selection Method 42
6.3.2 Identifying the User 44
6.3.3 Recommendations 44

7. CONCLUSION 46

8. REFERENCES 48

APPENDIX A: MENTIONED SYSTEMS 57

Acknowledgements

This report was part of a collaboration between DST and CSIRO. The project team at DST Group would like to thank Dr Cécile Paris, Science Leader at CSIRO Data61, for her valued and significant contribution to this report and the Consensus project. The author extends the same appreciation to Dr Nathalie Colineau, Dr Adam Saulwick, and Dr Kerry Trentelman who along with Dr Paris provided the detailed comments which have helped the author substantially improve this paper, and for their support through all stages of the report writing process.

Additionally, the author was part of a student placement from Swinburne University of Technology (Industry-Based Learning) to DST Group (Industry Experience Placement) in Edinburgh. He would also like to thank Dr Chris McCarthy for his provision of support as academic supervisor during the placement, along with those from both institutions who facilitated the initiation and sustainment of the placement.

Many of the bibliographic references that formed part of the literature survey could not have been acquired if not for the efforts of Ursula Amato—liaison librarian at DST Group—and Tristian Kemp—library collection specialist at CSIRO Library Services. We thank them both for their aid in searching and gathering the literature resources used to begin the literature survey that preceded this report.

Without the support and significant contribution of those above, this report could not have been possible.

The funding for the 2015-16 DST Group-CSIRO collaboration under which this report was written was provided under the Trusted Autonomous Systems Strategic Research Initiative ('Tyche' program).

Initialisms

ASR	Automatic Speech Recogniser
BDI	Beliefs, Desires, and Intentions
BN	Bayesian Network
CNL	Controlled Natural Language
CSIRO	Commonwealth Scientific and Industrial Research Organisation
DM	Dialogue Manager
DST	Defence Science and Technology
ECA	Embodied Conversational Agent
HMM	Hidden Markov Model
ISR	Intelligence, Surveillance, and Reconnaissance
ISU	Information State Update
MC	Markov Chain
MDP	Markov Decision Process
ML	Machine Learning
NLG	Natural Language Generator
NLU	Natural Language Understander
POMDP	Partially Observable Markov Decision Process
SDS	Spoken Dialogue System
SSJ	Sacks-Schegloff-Jefferson
TCU	Turn Constructional Unit
TRP	Transition Relevance Place
TTS	Text-To-Speech synthesiser

Glossary

Action selection The action of choosing the most appropriate response or action, given a user's input or the state of the dialogue.

Agent An abstraction of a software or human interlocutor, capable of individual and collaborative reasoning and communication. Agents are generally typified by an area of specialisation—in which that agent is capable—and areas in which it is not, hence necessitating inter-communication between agents to share knowledge.

Agent-based System architecture characterised by distributed processing and communication through the use of many cooperating software agents. Such systems decompose (or 'share') tasks they are assigned amongst their agents who are expected to complete portions of the task most relevant to their area of expertise.

Anaphora According to Jurafsky and Martin (2009), are references to entities which have been introduced earlier in the discourse. For example, 'her' in place of 'Jennifer'.

Automated planning The ability to generate a sequence of actions (a strategy), which may be composed of sub-actions and procedures, which the system believes will lead to a desired end state.

Bayesian network According to Russell and Norvig (2010), are data structures (in the form of directed graphs) that represent the dependencies among variables in any full joint probability distribution. Each node in the network represents an event and its conditional probability distribution—the quantification of the node's parents affecting the presence of that event.

Chat-bot A type of dialogue system primarily concerned with free, unrestrained conversation.

Context (in an SDS) Any kind of supporting information and knowledge to aid in the interpretation of dialogue or of user intentions and goals.

Contextual interpretation Determining the meaning of an utterance based upon its context—for dialogue systems this means the recent dialogue and other supporting evidence or characteristics.

Controlled natural language A subset of natural language with a restricted grammar and vocabulary.

Cost The resources lost or punishment incurred by the system, immediately or in the long term, when it undertakes a particular course of action. In spoken dialogue systems, cost can refer to time taken to complete a task, number of repetitions, or other metrics.

Deixis Expressions which require a point of reference to interpret (such as observing a pointing gesture) and can be categorised as: spatial (e.g., 'there', 'here'), temporal (e.g., 'before that', 'then', 'afterwards'), and interpersonal (e.g., 'those guys').

Dialogue context Information mentioned in or a characteristic part of the dialogue that is used to interpret the utterances in that dialogue.

Dialogue manager A component of an SDS responsible for the selection of actions and responses to user queries and other input, coordinating dialogue flow, error handling, and other high-level functions to facilitate these core capabilities.

Dialogue state A single snapshot or slice of the overall dialogue that captures the information, context, and knowledge that is intrinsic to the dialogue at a particular time in an interaction.

Domain knowledge Knowledge about the domain in which a system is placed (e.g., anatomy in medicine), including facts and relationships about objects and agents within that domain.

Embodied conversational agent An interface where the system resembles a human in both likeness and conversational ability (including nonverbal communication).

Finite-state machines Data structures that represent a series of events or inputs as discrete states and prescribed transitions between those states.

Frame-based Types of action selection concerned with the completion of frames—mandatory variables that must be filled by the user's utterances—where the choice of next action is usually dependent upon the frames missing or complete.

Grounding Actively ensuring that all participants have a mutual understanding about the discussion; in particular that they are speaking about the same topic.

Handcrafted action selection Action selection approaches that are characterised by a predominance of developer effort to specify the dialogue management capabilities of the dialogue manager.

Harel statechart A visual formalism that is functionally equivalent to a state diagram (Harel, 1987).

Hybrid action selection An SDS that utilises more than one approach in a combination to perform action selection in its dialogue management component.

Hypothesis An educated guess, often associated with a probability, that a system makes about the interpretation of the user's utterance or intentions.

Information-state update A method of dialogue management where information that constitutes the state of the dialogue at any given point is kept, and update rules are used to change (update) that state.

Machine learning A variety of techniques and algorithms used to detect and extrapolate patterns. For instance this may involve statistics or reinforcement and supervised learning algorithms to learn the most optimal decisions, given certain inputs.

Markovian model Refers to any of: Markov Chain, Hidden Markov Model, Markov Decision Process, and Partially Observable Markov Decision Process. All are stochastic models, but differ in their representation of the system state, depending upon: autonomy, and observability

Mixed-initiative dialogue The system provides general prompts for the user, but allows the user a degree of freedom with their responses.

Multi-agent A system architecture that is composed of multiple interacting software agents with robust methods of cooperation.

Multimodal Accepting (as input) or producing (as output) through a variety of channels: speech, gesture, keyboard and mouse, etc.

Natural language generation The ability to convert semantic representations of system responses into a series of natural language utterances.

Natural language understanding The ability to comprehend natural language inputs and convert them into semantic representations for use by a computer system.

Neural networks Forms of machine learning which utilise layers of interlinked neurons which activate based upon the weighted inputs they are given; these weights are learnt generally through backpropagation.

Non-task-based system A system whose purpose is to interact freely with the user and where an objective is not necessarily present.

Nonverbal Methods of communication other than speech.

Offline algorithm Any algorithm which can only perform computation on a discrete dataset whilst the system is not running.

Online algorithm Any algorithm which can compute upon real-time data as the system is operating.

Plan-based A type of action selection approach concerned with the interpretation of the user's utterances as speech acts as a means to infer and plan to achieve their intentions and goals.

Reinforcement learning A machine learning technique where a system learns from a series of reinforcements—rewards or punishments.

Restrictive grammar A form of language model that is dialogue-state-dependent.

Reward The resources gained or other positive consequence incurred by the system, immediately or in the long term, when it undertakes a particular course of action.

Rule-based A type of action selection method using IF-THEN rules, which match generally to inputs or dialogue states and act according to those rules.

Semantic representation The conversion of a (controlled) natural language utterance into a form that encodes its syntactical meaning.

Situational awareness According to Endsley (1988), it is the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future.

Speech act theory Work by Searle (1969) in describing the performative functions that utterances serve in a dialogue.

Spoken dialogue system (SDS) A specialised form of computer system that operates as an interface between users and the application, using spoken natural language as the primary means of communication.

SSJ Model The Sacks-Schegloff-Jefferson model (Sacks et al., 1974) for describing turn-taking behaviour for communication between humans.

Supervised learning A machine learning technique where an agent/system observes some example input-output pairs and learns a function that maps from input to output.

System-initiative dialogue The system directs rather than prompts, and the user is restricted to actions and utterances the system allows.

Task-based system A system with a clearly defined objective to achieve, and interaction with the user is a means to that end.

Text-to-speech synthesis The ability to convert natural language (textual) input into a natural language (verbal) speech output.

Transition relevance place According to Kronlid (2006): a place (in dialogue) where speaker-change is possible or preferred

Turn constructional unit According to Kronlid (2006): a phrase, clause, sentence, or word with a predictable end; an utterance

Turn-taking A fundamental concept in dialogue wherein each participant exchanges utterances (input to the conversation) in a particular pattern or sequence dictated by the needs of each participant and the structure of the dialogue interaction.

User model Knowledge kept of the user (or a group of users) for improvement and tailoring (of dialogue); the types of knowledge kept is developer-decided.

User-initiative dialogue Dialogue whereby the user is afforded maximal freedoms to direct the conversation and allowed inputs are unrestricted

Utility The benefit or gain (positive or negative) for a system when it takes a particular action; often domain-specific and defined by the designers

Wizard-of-Oz experiment A method of gathering user data where a human researcher plays the role of the system, interacting with human participants (the users)

1. Introduction

The motivation behind an investigation into spoken dialogue systems lies with several key guidance documents that lead the strategic directions for Australian Defence and National Security. For example the 2016 *Defence White Paper* (2016), the *First Principles Review* (2015), and the *2013 Australian National Security Strategy* (2013) all articulate the importance of a modernised, enhanced situational awareness capability which can anticipate threats and protect Australia's interests.

As situational awareness systems have become exceedingly complex, the requirement for bridging the gap between information the system knows and what the human user is capable of understanding is becoming ever more critical. The goal of simplifying the interaction process is to bring the system's knowledge and understanding up to the comprehension and language of the user.

A typical design choice of complex information systems is to require the user to learn and adopt the system's interface language, which is unnatural and requires specialised training. This forces the user to formulate their information requirements (inputs or queries) in terms of cumbersome, low-level schemas. We argue that it should be the role of enhanced situational awareness systems to provide users with the capacity to articulate their information requirements at a high level which is natural to humans. By providing complex systems with the capacity to enter into a natural dialogue with humans, we believe such systems have the potential to infer the high-level objectives of the user and produce their responses with an appropriate level of abstraction. In contrast, systems that are limited to low-level structured queries will only respond directly to those queries without regard to the broader context in which they were posed.

The goal of complex situational awareness systems, with their large data stores of knowledge, is to impart that knowledge effectively and efficiently to the user. We believe that a spoken dialogue interface to these systems allows the user to articulate their goals and objectives in a natural way, such that they are unburdened by the task of having to translate themselves to suit the system. With a natural language interface to communicate with a situational awareness system, the user is able to engage in knowledge enhancement through requesting information, clarification, negotiating, informing, and summarising. These are common dialogue strategies used between humans, and so by providing automated systems with this capability we give the humans the potential to gain a deeper understanding of complex situations.

This report aims to provide insight into the key issues and design considerations with regard to spoken dialogue systems, and in particular, dialogue management. It is hoped that it will serve as the basis for further research into the field, and promote further informed development within Defence. In order to produce a concise and relevant report, several decisions had to be made pertaining to its scope and coverage. This report was not designed to be a rigorous examination of the technical details of implementation and/or a thorough analysis of the theory surrounding dialogue systems—but instead it provides a brief review of some of the important concepts pertaining to spoken dialogue systems and dialogue management.

2. Background

2.1 Consensus

Consensus (Lambert et al., 2015) is a semantic-based high-level information fusion demonstrator developed at the Australian Department of Defence's Science and Technology Group[1]. The vision of the Consensus program is to provide intelligence analysts with a trusted, automated partner who can deliver in real-time deep situation awareness by enhancing and augmenting all-source intelligence analysis through semantic information fusion, automated reasoning and natural language question answering.

Consensus is currently able to ingest controlled natural language input from a user, in either spoken or textual form, and transform that input into a semantic format amenable to automatic inference. It is then able to perform automated logic-based and other inferences over all combined source data; this inferencing can be performed at all levels of the information fusion stack: sub-object, object, situation and impact assessments (White, 1988). This semantic-based information fusion capability is a cornerstone Consensus capability. Finally, the user is able to interrogate the system's knowledge base with queries; these are adequately answered if it is able to reach a logical conclusion through inferencing.

The interaction paradigm currently utilised by Consensus is a question-answer interface and does not resemble any form of natural dialogue. The user is restricted to a *controlled* natural language, which limits lexical and syntactical freedoms, and is unable to engage in multi-turn and lasting conversations—thus the interaction may be termed stateless or non-persistent. It is desired that a dialogue management interface be implemented to solve these issues present in the current iteration of Consensus.

2.2 Requirements

As previously noted, Consensus' current dialogue capabilities are limited, though our aspirations of the system's future are high: a system able to engage in naturalistic, multimodal dialogue with a plurality of users, and possibly a plurality of software agents. In order to achieve this, however, a list of concrete individual dialogue capabilities had to be devised such that the responsibilities of such a component could be scrutinised. As of the time of writing, the capabilities are still being defined, but they include specifications of: interaction length, number of participants, communicative acts, models of agents and users, and others.

The development of a spoken dialogue system is unavoidably a complex task from conception to implementation, and so choices were made regarding the prioritisation of

[1] Specifically, Consensus is a project being developed in the Language Technology Fusion Group within the Intelligence Analytics Major Science/Technology Capability, within the National Security & Intelligence, Surveillance, and Reconnaissance Division of the Defence Science and Technology Group—a group of the Department of Defence (Australia).

capabilities. These decisions have acknowledged that there exists a rough timeline wherein certain capabilities precede others. We recognise that such decisions have influenced this report, so that we have extensive coverage in certain areas whilst relatively brief analysis in others, as we focus on the most relevant characteristics—that is, with regard to time and requirements.

Some of the most important characteristics intended for a spoken interface within Consensus can be split into three categories: compatibility, extensibility, and trustworthiness. The first, compatibility, is a practical consideration and more of a desirable characteristic than a strict requirement—although it will impact the feasibility of implementation. Certain approaches amenable with Consensus' architecture and processes are discussed in detail in a later section (§6.3). Extensibility is similar to compatibility in that it is desirable, although it does have a basis in the literature. Cross-domain interoperability is a concern for dialogue systems whose inner workings have been specifically designed within a particular domain. This is of large concern to a Defence project which could be licensed to any third-party, and/or ported to a completely different scenario. Lastly, trustworthiness has been identified as an important requirement for Consensus as it is for human-computer interaction more generally. Some features—such as responsiveness, reliability, and lifelikeness—will improve an agent's similarity to a human interlocutor, and thus will allow it to develop a long-lasting rapport with the user. A rapport is beneficial for both parties: the user feels as though they can rely on the agent's output, and the system can rely on the user to interact with it more often.

We believe that in commencing the development of a new functionality to any complex software system—a dialogue management interface to Consensus—the best practice is to be aware of research efforts that have already been conducted in describing such functionality at either a theoretical or technical level. This report was devised as a means to: survey historical and extant literature within the domain of spoken dialogue systems and dialogue management, identify key themes in that literature, present some systems that have already been developed, and make considerations that may be of importance to any future efforts towards Consensus' dialogue capabilities.

3. Methodology

This section outlines the search strategy and selection criteria adopted for this literature review, as well as the considerations taken into account throughout the research and development of this report.

3.1 Survey Process

This research into dialogue management is part of a joint collaboration between DST Group and CSIRO. The project team comprised: Deeno Burgan, Dr Colineau, Dr Paris, Dr Saulwick, and Dr Trentelman.

Papers would be sought by searching through three key resources: the DST Group research library, the CSIRO research library, and open access web resources such as Google Scholar and others. Initial searching comprised keyword phrases which were deemed relevant to the domain of dialogue management, but which were not too narrow in scope to avoid missing relevant results. Additionally, in order to capture the work of authors who may have not appeared in the initial search, specific journals and conferences of relevance to this domain were targeted with papers and their authors included in the survey.

It was recognised that spoken dialogue systems have been an ongoing research topic for many decades. In order to produce a concise review of issues pertinent to this domain, it was decided to restrict much of the survey to literature produced since the 2000s, however, it was realised that such a decision would miss some of the most influential (and in some cases definitive) authors—and excluding their efforts would negatively impact on the writing of an informed review.

Hence, a balance was struck between old and new; seminal and notable papers, even of greater age, would be included due to their impact on the landscape of dialogue systems research. Recent papers would also be considered if they were likely to describe systems and theories that were much relevant or extant.

Once a large repository of citations and abstracts had been collected through querying of the available research libraries, naturally a number of irrelevant pieces—such as those pertinent to a completely different domain or field—were identified. To remedy this, selection criteria (see §3.2 below) were established and papers selected through a series of iterative processes. The aim was to reduce the large list of approximately 350 papers to a smaller, manageable set that could be analysed in greater detail. During these stages in the survey, the broad areas of focus had not been established, and so the research encompassed the broad categories of dialogue systems and dialogue management.

After the papers gathered via the initial trawl had been reviewed, broad surveying stopped, and then a refined process of selection began, again detailed in §3.2. By this point, key themes in the literature were identified, and papers kept could be structured accorded to themes.

Additional literature surveys were made as a focussed attempt to fill gaps in knowledge. This included following the chains of referencing present within papers in order to discover additional material—in particular, seminal papers that set the stage for a particular area of interest in dialogue systems, or that provided definitive definitions for those areas.

3.2 Selection Criteria

Given that the focus of this report is on dialogue management, and spoken dialogue systems generally, it was reasoned that the rejection of literature relevant only to other fields—those which did not cover spoken dialogue systems or dialogue management specifically—would be reasonable. In the course of surveying the literature, it was discovered that dialogue systems had found implementation within numerous domains (e.g., assisting patients with post-traumatic stress disorder and medical diagnosis); literature of this kind was retained for further analysis. Research papers which contained no mention of dialogue systems or dialogue management were removed from the survey list. These judgements were made on the basis of abstracts or titles of the papers in the absence of the former. Decisions as to whether to keep or discard a paper were also made with regard to the overall requirements of Consensus as described in §2.2. The project team met regularly to discuss papers and make collective decisions.

Once a smaller repository of papers had been created, a refined analysis of the literature occurred. The project team read these papers in their entirety, making a judgement about the utility and relevance of any system or theory described therein.

In the latter stages of the survey, tailored searching occurred in order to fill gaps in knowledge not met by earlier iterations. Papers were found using keywords that were specific to the area of focus for a particular section (e.g., logic and reasoning pertaining to agent-based systems) or within domains of expertise of the individual reviewers (e.g., user modelling). Given the concentrated nature of this process, it was reasoned that each paper gathered in this way would be of sufficient rigour.

3.3 Report Structure

The structure of this report reflects what was identified in the literature as being areas of interest for dialogue systems. It was noted that each topic could serve as a main research interest of an entire report by itself, and so for brevity a general approach to this review has been taken. We provide a light treatment of each issue with a definition and description of how it pertains to dialogue management with a brief analysis benefits and disadvantages.

The remainder of this paper has been divided according to two related concepts: spoken dialogue systems (§4), and dialogue management (§5). In the former we provide a background on spoken dialogue systems: their typical constituents, development of multimodal capabilities, and finish with a history of some key dialogue systems. In the latter part we delve into issues pertinent to dialogue management—a critically important feature of dialogue systems—covering: action selection, multi-agent architectures, understanding the user, and the ability to handle errors. We finalise this report with a two-part discussion (§6) and conclusive thoughts (§7).

4. Spoken Dialogue Systems

4.1 Definition

A spoken dialogue system (SDS) is a specialised form of computer system that operates as an interface between users and the application, using spoken natural language as the primary means of communication (McTear, 2002). Flycht-Eriksson (2001, p. 2) defines dialogue systems as computer systems that 'interact with users, utilising connected natural language dialogue, where the use of language need not consist of predefined commands. It is claimed that spoken conversation with such systems—in a manner similar to that of human-human dialogues—allows for a natural, intuitive, robust, and efficient means for interaction (Skantze, 2007). Such conversational systems have been in research and development for several decades (McTear, 2002). As a summary of their function, they typically accept speech input from users, recognise and understand the meaning of the input, and finally respond appropriately. SDSs are typically useful in assisting users interface with complex task-based systems where it is beneficial to offer a user-centric interface (Lee et al., 2010) as opposed to having to learn interfacing languages that coerce the user to the system's representation paradigms.

We concede that the definitions given above may not clearly illustrate what spoken dialogue systems are, and in remedy we delve into some of the components which form their constituents.

4.2 Components

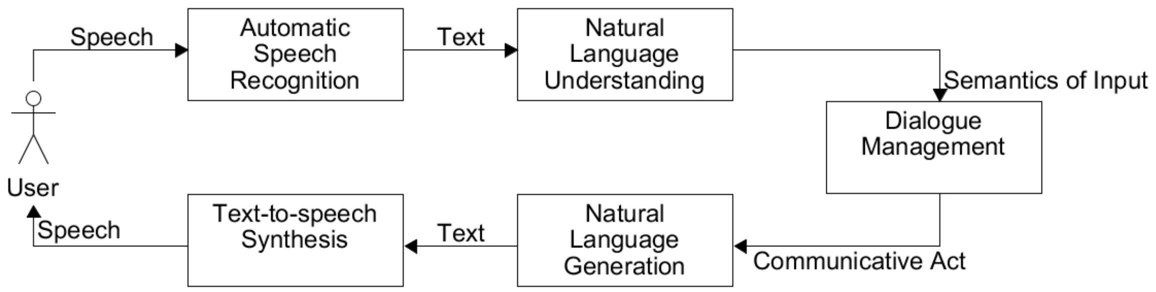

Figure 1 - Spoken dialogue system input/output flow

SDSs are often represented, designed, and developed as a process flow between several communicating components as shown in Figure 1. In our diagram, boxes represent key processing stages and arrows link one stage to another—arrow text highlights the form of data being sent between processes.

4.2.1 Automatic Speech Recogniser

The beginning of the process starts with the automatic speech recogniser (ASR) whose role is to recognise the sounds the user is making, a sequence of acoustic-phonetic parameters, and convert that into the string of words that have been uttered (McTear, 2002). Due to the inherent uncertainty of speech detection in these systems, this conversion is usually only a hypothesis, perhaps one of many, and is usually associated with a confidence score (Skantze, 2007). The range of input words and sequences recognised by the ASR component is dependent on the grammars and language models it is given. Jurafsky and Martin (2009) distinguish between restrictive grammars that are hand-written and tailored to certain domains, and others that are of a general and probabilistic nature. In dialogue systems, the ASR may utilise learning techniques to recognise and adapt to the speaker who is interacting with it, improving speaker recognition.

4.2.2 Natural Language Understanding

The ASR feeds its hypothesis of the user's input, now transformed into textual form, to the natural language understander (NLU) which produces a 'semantic representation that is appropriate' (Jurafsky & Martin, 2009, p. 858); its purpose is also described as the capacity to extract the meaning of an utterance (Skantze, 2007). A major task of the NLU is that of parsing, taking a string of words and producing a linguistic structure for the utterance (Jurafsky & Martin, 2009). The method by which an NLU parses input is implementation-dependent and ranges from the use of context-free grammars, pattern matching, or even data-driven approaches (Skantze, 2007). The particular representation adopted by the NLU should be able to be understood and used by the dialogue manager (Lee et al., 2010).

4.2.3 Dialogue Manager

Following the NLU in the SDS process is the dialogue manager (DM), an important module whose purpose is to coordinate the flow of the dialogue and communicate with other sub-systems and components. LuperFoy et al. (1998, p. 794) characterise the DM as an 'oversight module' that 'facilitates the interaction between dialogue participants'.

In order to do this it must receive the user's input from the NLU and produce the system's responses at a concept level to the natural language generator. Which response it chooses will depend on the strategy that has been chosen; another facet of responsibility attributed to the DM. Strategies are related to the keeping of a conversation's state and the ability to model the dialogue structure beyond that of a single utterance (Jurafsky & Martin, 2009, p. 863).

The importance of dialogue management is made by Larsson (2002, p. 2) who states that, in order for SDSs to achieve flexible dialogues with users, its implementation must be based upon 'reasonable theories of dialogue modelling and dialogue management'.

Skantze (2007) believes that the tasks of the DM may be categorised into three groups:

- contextual interpretation—the ability to resolve ambiguous and referring expressions
- domain knowledge management—ability to reason about the domain and access information sources
- action selection—deciding what to do next.

Contextual interpretation usually requires keeping some form of dialogue context which can be used to resolve anaphora[2] and deixis[3] (McTear, 2002); and this context may have a number of constituents: dialogue history, task records, and other models (e.g. user models), which all can be used as knowledge sources and together may be collectively referred to as a 'dialogue model' (McTear, 2002). We discuss further the issue of context, with regard to dialogue management, in §4.3.

Skantze (2007) uses the definition of knowledge sources (pertaining to a DM) given by Flycht-Eriksson (2001) as the ability of the DM to reason about the domain in which it is placed; part of that involves the representation it keeps about the world. Flycht-Eriksson also claims that domain knowledge kept by the DM can be used in its other tasks.

The third responsibility of the DM is the choosing of the dialogue system's next action to take; it develops or selects strategies that allow the DM to decide what to say or do given the current and previous state of affairs.

The way in which a DM chooses its actions also has an effect on who has initiative through the conversation. In spoken dialogue systems, initiative refers to the participant who has the control of the dialogue at any given time—they are able to choose how much to say and what to talk about, and so on. At one extreme, there exist system-initiative dialogue systems, where it is the role of the spoken dialogue system to lead the user through the conversation, prompting them at every stage; at the other end are user-initiative dialogue systems that allow the user complete control—these include systems such as chat-bots, whose purpose is simply to provide conversation, or some task-oriented systems that allow flexibility in how the user approaches a task; finally, as a compromise, there are mixed-initiative systems which have an overall end-goal that must be achieved, orchestrated by the dialogue system, though it will afford the user a larger degree of freedom in how they choose to respond. A number of methodologies to select actions have been proposed in the literature, and they are presented in §5.1. They include methodologies such as finite-state machines, used in early SDSs, to machine learning techniques adopted in recent systems.

LuperFoy et al. (1998, p. 795) list five key capabilities that a dialogue manager fulfils:

> 1 Supports [a] mixed-initiative system by fielding spontaneous input from either participant and routing it to the appropriate components.
>
> 2 Supports non-linguistic dialogue "events"[4] by accepting them and routing them to the Context Tracker (below)[5].

[2] Anaphoric references are expressions that stand in reference to objects already mentioned earlier (e.g., 'her' in place of 'Jennifer').

[3] Expressions which require a point of reference to interpret (such as observing a pointing gesture) and can be categorised as: spatial (e.g., 'there', 'here'), temporal (e.g., 'before that', 'then', 'afterwards'), and interpersonal (e.g., 'those guys').

[4] LuperFoy et al. refer to a specific form of multimodal interaction, the use of gestures, when they speak of non-linguistic dialogue events.

[5] We briefly return to LuperFoy et al.'s treatment of Context Tracking in §5.1. The bracketed 'below' is a result of directly quoting their work and has no meaning in this report.

3 Increases overall system performance. For example, awareness of system output allows the Dialogue Manager to predict user input, boosting speech recognition accuracy. Similarly, if the back-end introduces a new word into the discourse, the Dialogue Manager can request the speech recognizer to add it to its vocabulary for later recognition.

4 Supports meta-dialogues between the dialogue system itself and either participant. An example might be a participant's questions about the status of the dialogue system.

5 Acts as a central point for dialogue troubleshooting, after (Duff et al. 1996). If any component has insufficient input to perform its task, it can alert the Dialogue Manager, which can then reconsult a previously invoked component for different output.

The above list presents no new information about the role the DM has in an SDS, as has already been presented in this sub-section. Although one could draw some small distinctions, the key point to be made is that the end state is always the same: the DM serves to create a robust link between the user's utterances and the system's actions (which themselves may be utterances), and keeps track of information that it utilises to reach that goal.

The key expected outcome of the DM according to Skantze (2007) is a semantic representation of a communicative act, although Bohus and Rudnicky (2009, p. 333) refer to a system action 'in the form of a semantic output' and LuperFoy et al. (1998) attribute 'various' output formats to the DM. The terms communicative act and semantic representation are both used in the literature but we believe they should be seen as completely separate notions. The former is an intention to do something—ask, tell, persuade and so forth—whereas the latter constrains this intention to a concrete format (e.g., 'please tell me your name').

We believe that the DM exports an intention: 'I need to *ask* the *user* for their *name*'. That example is in natural language but the italics help us imagine how the system might encode such an intention: *ASK(USER, NAME)*. This second example is much clearer as it no longer confounds the issue of language generation which is the responsibility of the following component.

4.2.4 Natural Language Generation

The natural language generator (NLG) receives the specification of a communicative act from the DM and generates a matching textual representation. Jurafsky and Martin (2009, p. 861) define two functions that the NLG must perform: content planning and language generation, but acknowledge that the former can be attributed to the DM instead. Content planning involves deciding the semantic[6] and pragmatic[7] content of a speech act, what the

[6] Semantic content, in the linguistic community, is the syntax used in an utterance (e.g., its structure) and what lexical items it contains.

[7] Pragmatic speech content is dependent upon the context in which the utterance is spoken which can change the implication of the utterance altogether (i.e. in sense-making and function).

system intends to convey to the user. Language generation in contrast is the realisation of the meaning by 'choosing the syntactic structures and words needed to express the meaning'.

The definitions given above may appear to confound the distinction between the NLG and the DM, as both inevitably add to the final system output. In order to rectify the confusion we provide an abstract example illustrating how a response is generated from the efforts of both components:

1. The DM in a battlefield-scenario SDS decides, perhaps in following a dialogue strategy, that during the next turn it must give the user an update of their situation on the ground.
2. The DM sends the conceptual representation of a communicative act, that it intends to fulfil its goal of informing the user.
3. The NLG, having received the communicative act, expands the act into language by forming a semantic representation: 'Your position is at … and you are near town …' Here, it is the responsibility of the NLG to decide what information is included in the response, and how it should be presented in language.
4. The textual response is sent to the speech synthesiser for production and output to the user.

In the above example, the DM has decided the end state it intends to achieve through communication (provide an update on the user's situation), but it is the NLG that decides how to get there by developing the language and content that will be used.

4.2.5 Text-to-speech

Text-to-speech (TTS) is the module responsible for conveying the output of the response generated in the NLG to the user through synthesised speech. This involves the translation of the response into a spoken form (McTear, 2002) which can be achieved through a number of different methods, the simplest of which being the use of pre-recorded (or 'canned') speech which are typically utilised in systems whose outputs are expected to be given in either precise or predefined formats. Both McTear (2002) and Skantze (2007) characterise TTS in terms of two processes: text analysis, a mapping of the text to their matching phoneme representations—including an analysis of linguistic structure and prosodic mark-up, and speech generation—whereby the annotated speech act is finally vocalised to the user.

4.3 Multimodal Interaction

Spoken dialogue systems, true to their name, are usually designed with a clear focus on natural language communication—often verbal, but in some cases text-based. However, some other research has been conducted into endowing SDSs with additional modalities—ways of accepting input through a variety of different senses, and communicating through alternative outputs.

It is claimed that by offering modalities other than natural language, the user is able to interact with the system more naturally by utilising nonverbal communication techniques used by human interlocutors in human-human conversations (Cassell et al., 2000). Additionally, certain kinds of information may be more easily conveyed across these nonverbal channels than what could be achieved by vocalisation. An SDS able to interpret a user's bodily gestures could infer their level of attentiveness and valence during the interaction (Schröder, 2010), a characteristic that is not often conveyed in speech. By linking verbal and nonverbal aspects of communication, it is believed the overall conversational capabilities of virtual agents can be improved (Riviere et al., 2011).

Including multimodal capabilities in an SDS comes with its set of challenges; a key one is the increase in decision-making required in order to produce a communicative act expressed across multiple modalities. Additional consideration must be made to coordinate each modality to ensure that information is conveyed uniformly and is not presented in a way that stimulates sensory overload.

In a similar way to production, the SDS must also effectively interpret user input across different modalities—an involved process that entails making sense of the user's communicative act from several perspectives. One of the biggest challenges is ensuring that a correct overall user intention can be gleamed from several disparate modes of input. If the system misinterprets a hand gesture made by the user, but correctly identifies their speech or text input, the DM may still make incorrect assumptions about the user's intent.

It may be argued that by considering only textual contributions from the user, a developer is able to simplify the aspects of comprehension by avoiding the issue of error-prone and fuzzy inputs. This is a valid decision, but one that should be taken with caution; by ignoring gesture and various kinds of body language and intonation in speech the system is at a significant loss by not exploiting these rich knowledge sources.

We contend that alternate modalities serve to bolster the baseline understanding of the user that natural language modalities afford, as well as offer the user a freedom in expressing their needs and goals.

The implementation and use of natural language modalities—text and speech—will not be mentioned here as it is a topic that is well documented in the literature (Jurafsky & Martin, 2009; Leuski & Traum, 2008; Shang et al., 2015; Sordoni et al., 2015; Van Noord et al., 1999). We instead focus upon alternative modalities whose inclusion in a spoken dialogue system is perhaps not as obvious. Furthermore, the discussion of how such modalities are handled (i.e., the use of natural language understanding and natural language generation) will likewise not be covered due to their independence from dialogue management, and sufficient knowledge of those topics has already been provided in this review.

4.3.1 Embodied Conversational Agents

Cassell (2001, p. 67) provides a useful definition for an embodied conversational agent (ECA):

> …an interface in which the system is represented as a person, information is conveyed to human users by multiple modalities such as voice and hand

> gestures, and the internal representation is modality independent and both propositional and non-propositional.

Cassell's use of the term 'propositional' refers to information intended to convey some knowledge of the world, which may be carried out through a modality. Non-propositional behaviours are not explicitly defined in the 2001 article, although in a prior article she highlights a division between propositional and interactional contributions (Cassell, 2000). Interactional contributions are defined therein as 'cues regulating the conversational process and includes a range of nonverbal behaviours', going on to say that 'interactional discourse functions are responsible for creating and maintaining an open channel of communication between participants'.

The commonly cited benefits of embodied conversational agents for interacting with users are:

- Speech is the most effective means of conversation for humans, and thus ECAs are 'powerful ways' for users to interact with their computers (Cassell et al., 2000).
- Human-like conversation with the ECA provides a natural and intuitive interface to a system (Pelachaud, 2005).
- recognising and producing nonverbal behaviours creates an immersive experience for the user (Lee & Marsella, 2006).
- Dialogue acts can be produced though certain modalities without interrupting others such as head nodding whilst the user is speaking (Cassell et al., 2000).

The use of ECAs comes with several considerations, some challenges to overcome as well as intrinsic weaknesses introduced by the use of characters acting as a second (human-like) interlocutor to a conversation:

- It is not guaranteed that the user will have a positive attitude toward it, and this can affect how the user interacts and behaves during conversation with the ECA (Novielli et al., 2010).
- The development of a true ECA must include the ability to understand and generate nonverbal behaviours; a non-trivial task that introduces additional complexity to the system.

The benefits to be gained from ECAs outweigh these apparent disadvantages—which are mostly challenges that can be overcome. Users may eventually grow accustomed to the paradigm of speaking to an agent, either through training schemes or mere exposure to the system. Additionally the complexity of development is a challenge that may be handled, though perhaps not solved, with satisfactory infrastructure modelling and sufficient decoupling of system components. Overall it appears that ECAs have much to offer to spoken dialogue systems, including the ability to handle nonverbal interactions—akin to the abilities employed by humans when communicating with each other.

4.3.2 Visual

Visual systems feature prominently in the ECA literature, using various visual recognition techniques in order to make various assumptions about human interlocutors. The majority of participants in human-human conversations have access to visual cues and may use these to infer a lot of information about the nature of the current dialogue and others involved. In discussing their use of engagement-aware behaviours—the kind derived from acknowledging visual cues—Xu et al. (2013, p. 2240) state that they had a 'positive effect on the evaluation of the agent's humanness, intelligence, likeability, and overall user satisfaction'. Thus by picking up and acting upon visual information data during conversation, SDSs may benefit by inching closer to realistic interactions as they further approximate the perceptive abilities of humans in conversation. Whilst the modality is 'visual', a number of features or facets of interaction can be discerned: eye contact, hand gestures, facial expressions, and others.

An effort made by Moussa et al. (2010) utilises nonverbal behaviour annotations, which they had gathered from an empirical study they conducted ,to develop a virtual tutor (in an embodied conversational agent) that is able to exhibit those behaviours in a frequency similar to that observed in the experiment. Eye contact was especially prominent in this research and was linked to specific events throughout a tutoring session. A summary is given by Lee and Marsella (2006) for the study of nonverbal behaviours during face-to-face communication, including the importance of head movements—as several kinds of functionalities are possible for head movements. Clearly, a number of nonverbal cues and actions have a great effect on how a particular speaker is interpreted. It is thus no surprise that the research literature is taking advantage of this by incorporating visual systems into conversational agents.

An ECA by Xu et al. (2013) records several visual cues—lip motion, facial expressions, eye contact and gaze, and body distance, motion, and others— in order to determine a user's saliency (their attention toward the agent) and engagement intentions (to obtain or release speaking turns). Their system, a robotic conversational agent, attempts to recognise the engagement of users in a multiparty conversation. The intentions of each participant along with the floor state (who currently has initiative) affects the behaviour of the agent, deciding what it would say and to whom. The use of multimodal inputs and their subsequent classification into intents led to the turn-taking behaviour taken by the ECA.

The use of gestures combined with speech has been described as integral to conveying and understanding propositional content (Cassell et al., 2000), where it is the combination that expresses the whole underlying representation. The use of nonverbal cues also allows assumptions to be made of conversational partners, such as in the system by Nass et al. (2000) where the individual's gestures can be indicative of their extraversion or introversion. Such characteristics can form the basis of a user model or motivate the use of an alternative dialogue strategy to cater for users based on what the system has observed.

4.3.3 Emotion

The rationale for considering emotion in an SDS—sometimes referred to as 'affective reasoning'—is to create natural and intuitive interactions between humans and machines (Schröder, 2010). In particular, Riviere et al. (2011, p. 2) consider that 'an agent able to not only verbally express [a] sentence but also multimodally express the underlying emotions will be more sincere and believable to the user.'

The ECA presented by Smith et al. (2011) is capable of engaging with the user in free (although domain constrained) conversation with the premise of discussing a user's day at the office, a domain chosen due to the likelihood that it would contain affective content. The purpose of the ECA's responses is to positively affect the user's attitude by replying to their situation in certain ways depending on how they themselves describe their situation; the responses range from reassurance, advice, comfort, or warnings. Crucial to generating a relevant response is the appraisal of the nature of the event the user describes as well as the response they have to that event— if an event has deteriorated their situation at work, then the system may reassure them or provide useful advice. The system's architecture contains several modules including an 'affective strategy module' used for generating affect responses.

A second system with an emotion aspect at its core is SEMAINE (Sustained Emotionally coloured Machine-human Interaction using Nonverbal Expression) (Schröder, 2010) whose aims are to create multimodal dialogues with emphasis on nonverbal detection and production; in regard to the latter, the embodied agent is able to produce facial expressions relevant to the conversation. The SEMAINE system includes a number of personalities that represent a particular emotion, serving to affect the tone of their responses; a demonstrative conversation with one of these personas, Obadiah, is viewable on YouTube[8].

Emotions can clearly change the way spoken dialogue systems speak with users; changing a monotonous conversation into an interaction where users have a 'fruitful and enjoyable experience' with the system (Pelachaud, 2005, p. 1). Schröder (2010) claims that by utilising such systems, there is the potential to close a divide that exists between expert users and those who feel 'helpless in front of increasingly complex technology'.

4.4 History of SDSs

One of the earliest systems, and one of the most known, was the chat-bot[9] ELIZA developed by Weizenbaum (1966) during the mid-1960s. It utilised primitive pattern matching techniques to respond to the user's statements, primarily in the role of a Rogerian psychologist. ELIZA received an input from the user, inspected it for keywords, and transformed the sentence to form an output based on rules associated with those keywords—the set of keywords and their respective rules constituted the script for the system. Despite these simplistic approaches, some users believed it to be significantly more intelligent and realistic than it actually was, giving rise to what would be termed the *ELIZA* effect (Hofstadter & Boden, 1995).

[8] https://www.youtube.com/watch?v=zruOPSSWVXw. Chatting with a Virtual Agent: The SEMAINE Project. Accessed 31 October 2016.

[9] The term chat-bot was not current at the time.

Two years after ELIZA, directory Stanley Kubrick and writer Arthur C. Clarke released the film *2001: A Space Odyssey* which depicted one of the most iconic characters in science-fiction and artificial intelligence (AI): HAL 9000. HAL exemplified a number of interesting traits of dialogue systems, just as it did for artificial intelligence—if one ignores its bout of murderous intent. For instance, its ability to comprehend the natural language sentences of crew members and respond in a human-like way is a milestone that is yet to be matched to such a degree with current systems. However, some aspects have been met with research and academic efforts since the movie's release. The history of speech recognition has shown improvement in accuracy of word detection, and text production systems (e.g., text-to-speech) are becoming increasingly robust. One particular characteristic of *HAL*, of importance to this paper, is its ability to converse freely with humans—not simply in a back-and-forth, question-answer paradigm that would have been typical of teletype systems of that era. This capability has not yet been achieved by any known system.

Since the appearance of ELIZA and the popularisation of AI—in part a courtesy of HAL—spoken dialogue systems became a topic of research interest for academia and industry. In order to provide a glimpse of the path that such research has taken, we present a brief chronological look at some of the various systems of note that have been observed through our survey, noting specialisations[10] of each.

We begin with the TRAINS project (Allen & Schubert, 1991), a plan-based dialogue system whose aims are to interact with the user in a mixed-initiative dialogue with robust natural language understanding, and produce real-time plan reasoning and execution. The domain for this system was in the management of cargo through movement of trains between destinations. *TRAINS* does not physically interact with the domain in order to cause changes, rather it formulates plans and monitors them—this includes interacting with agents, human or software, in order to find out information about the world.

Many early dialogue systems did not focus on natural and realistic (human-like) conversations, as they were designed simply as user interfaces to conduct actions within complex systems. SpeechActs (Yankelovich & Baatz, 1994) is one example of a system whose goal is to explore dialogue, but primarily from the point of view of speech recognition and processing; and very little was documented in the areas of dialogue management. To its credit SpeechActs included a discourse manager—perhaps a precursor to the DM component—but its services were limited: the management of user and system information—with regard to the blackboard architecture, prompting the resolution of simple ambiguities, and switching to other applications. The Philips automatic train timetable information system (Allen & Perrault, 1980) is another example, which claimed to have some unspecified dialogue management component amongst other planning subsystems.

AutoTutor (Graesser et al., 1998; Wiemer-Hastings et al., 1998) is an interesting project grounded within a pedagogical domain that uses dialogue to teach students. It is a primarily rule-based system that uses a 'curriculum script' that contains the various topics it is capable of discussing with a learner. Decisions about which topic to pursue are made

[10] Some dialogue systems have a particular focus, such as producing responses or producing mixed-initiative dialogues, and so are typically not 'all encompassing' in their operation.

via production rules that consider additional information regarding the learner's skill level, the needs and goals of both student and teacher, and other global variables. Its key research aims are to determine 'what the tutor should say next', referring to this as the 'conceptual content', and contrast this with the non-aim of 'how the tutor should say it'. We direct the motivated reader to a comprehensive review of its history and contributions, as articulated by Nye et al. (2014).

The CommandTalk system (Moore et al., 1997; Stent et al., 1999) makes use of a dialogue stack (composed of user-system discourse pairs—frames), semantic representations of user input and system responses, and is able to maintain context. Although it utilises finite-state machines for the handling of different kinds of conversations, it appears to be suitably robust for its application in serving as the spoken interface to a defence battle simulator.

In 2000, Larsson and Traum released a paper documenting the TrindiKit Dialogue Move Engine Toolkit Larsson and Traum (2000) which personified a new paradigm for dialogue management: the information-state update approach. The focus for this approach is the recognition of key characteristics of dialogue that change as the dialogue itself changes and, importantly, how they are changed. We return to information-state update later in §5.3.2.2, where we discuss it as a method of handcrafted action selection.

Another project we believe of importance to the history of SDSs is Carnegie Mellon University's Communicator (Rudnicky et al., 1999) the precursor to *Olympus* (Bohus et al., 2007) and its RavenClaw dialogue manager (Bohus & Rudnicky, 2003, 2009). RavenClaw opts for a frame-based—also referred to by the authors as agenda-based—approach to dialogue management, utilising hierarchically structured set of information frames that need to be filled from the user's utterances. The benefit of this system is that it results in a mixed-initiative dialogue that affords the user flexibility; useful for domains that need to elicit information to perform extended queries (e.g., booking systems).

Other approaches to the choosing of dialogue acts have utilised machine learning techniques (Lee et al., 2010; Lemon, 2011; Williams & Young, 2007; Young et al., 2007). The goal of these approaches is to achieve a degree of adaptation in dialogue management both to new situations and to new users (Papangelis et al., 2012a). Adaptation is one rationale, but another is error correction which can be difficult for human designers to anticipate. Machine learning systems should be able to undergo automated planning that decides the corrective measures to take that are the most useful in the long run (Williams & Young, 2007)—this usefulness is associated with utility and is intentionally left ambiguous for designers to define.

In a shift of focus, some dialogue systems have explored multimodal interaction—the use of embodied conversational agents (ECAs), gesture input/output, eye contact, and others. One such system is SEMAINE (Schröder, 2010), an affective-oriented (emotion) dialogue system, which produces facial signals during conversation and tracks the user's emotional state represented in an arousal-valence plane. This system is discussed later in §4.3.3.

Some proprietary systems have gained notoriety in recent times, perhaps due in part to the companies that have backed their development—and the resultant competition between them. Siri, Cortana, and Google Now are intelligent personal assistant software agents developed by Apple, Microsoft, and Google, respectively. We do not give an in-depth analysis of such systems here, as we focus on open and/or academic systems publicly described in the literature. We credit those systems for their robust natural language processing capabilities and impressive backend processing in order to answer direct user queries—even providing answers the user did not directly ask for. We must however conclude by stating that these systems do not, at time of writing, conduct dialogue management; they only operate in a question-answer paradigm and so are not relevant for our discussion here.

In this section we have had a brief look at a small minority of the SDSs that have been detailed in the literature we have surveyed. The theoretical and technical capabilities of such systems have clearly improved over time, and in this report we hope to expose some of those advances.

5. Dialogue Management

In the previous section we described what SDSs are, providing examples of systems that have been developed in the literature and the attention that has been paid to developing SDSs as ECAs—whose focus is to communicate with multimodal channels, similar to a human conversation partner. In explaining the components of which an SDS is comprised (§4.2) we made reference to the dialogue manager (DM) as being critical to choosing the system's responses and the overall direction of the dialogue. Thus in this section we expand upon the DM and the fundamental capabilities it must achieve to be useful within an SDS.

We begin with the notion of context in dialogue (§5.1), and how it may manifest early in speech recognition and even to the higher-level management of dialogue. Then we identify the importance of turn-taking strategies (§5.2) to decide the speaking order between interlocutors, and then the approaches a DM may use to decide what it says next (§5.3). We go on to highlight the usefulness of multi-agent architectures (§5.4) and how such agents might be composed in an SDS, followed by the importance of understanding the user (§5.5). We finish with a description of the kinds of errors (§5.6) that may arise through interaction with the user, and why it is important for a DM to handle these adequately.

5.1 Context

The use of the term context is often encountered within the setting of spoken dialogue systems, and its purpose differs depending upon which stage in the process it is gathered and applied. However, an overarching link may be drawn between these applications: they supplement the understanding of the user's communicative behaviours (Bunt, 1994).

At the beginning of a spoken dialogue system's process, context takes on the role of allowing the ASR to resolve ambiguities by using previous utterances or knowledge of the domain to improve hypotheses with low confidence scores. McTear (2002, p. 106) describes this potential in the context of a flight enquiry system that could discard certain input hypotheses if they are found to be 'contextually irrelevant' based on the domain. In that example the hypothesis 'what time does the white leaf' would be flagged as irrelevant due to its use of terms outside the flight domain and instead be left with hypotheses such as 'what time does the flight leave'. An effort made by Jonson (2006, p. 177) put into practice the usage of dialogue contexts to improve the hypotheses of an ASR, which would result in the betterment of dialogue flow as a result. Jonson found 'considerable ASR performance improvement' with the use of a trained classifier to assign categories to ASR hypotheses based on grounding categories.

Context tracking as described by LuperFoy et al. (1998, p. 795) is a component of discourse processing alongside dialogue management and pragmatic adaptation, although represented as following natural language processing. Used in this way, context allows for the resolution of dependent forms[11] present in the input and the ability to 'produce context-dependent forms for achieving natural output'. They perceive context tracking as an independent process whose inputs and outputs are logical forms, with dependent references resolved in the latter.

Rickel and Johnson (2000) experiment with a virtual embodied agent named STEVE (Soar Training Expert for Virtual Environments). They discuss the importance of maintaining a rich representation of context in order to facilitate coherent behaviour for a virtual agent in a dynamic environment (also virtual), distinguishing two kinds of context as pertinent to STEVE:

- task context
- dialogue context.

Within the dialogue context they associate the following information: current speaker, individual with task initiative, series of steps that have been executed, a representation of a discourse focus stack as described by Grosz and Sidner (1986), and keeps track of user requests. The last datum is of interest as it allows STEVE to maintain requests until such time as it can fulfil them (i.e., if the current task is of greater importance and must be completed first).

We summarise the importance of context by stating that is critical in aiding the understanding of the user; allowing the DM to bring additional information to its processing of input.

5.2 Turn-taking

In order to facilitate a dialogue beyond single-utterances, a DM must be able to decide at which point each interlocutor (the system or the user) gets to speak. Thus, turn-taking behaviour in an SDS involves a set of rules and procedures that allow separate agents to communicate efficiently by determining who is to stop speaking and who is to begin. It is

[11] Luperfoy et al. use the term *dependent forms* in referring to definite pronouns, demonstratives, indexicals, definite NPs, one-anaphora, and ellipsis.

concerned fundamentally with the process by which interlocutors exchange utterances and speaking turns.

The case for acceptable turn-taking strategies in an SDS is salient. It is the claim of Raux (2008) that a greater cost is associated with additional turns in an SDS than in human-human conversation—due to the greater disruption to dialogue. Kronlid (2006) sought to find turn-taking strategies as applicable to a multi-party scenario where the communication between agents is not constrained and thus crucial to maintain order. Turn-taking abilities are critical in negotiation interactions, as they can influence the establishment of rapport and solidarity, or expressing a position (DeVault et al., 2015). The need for turn-taking is summarised by Raux and Eskenazi (2012, p. 1) who state that 'in order to lead productive conversations, people need not only know what to say but also when to say it'. It is clear that an SDS, and the agents of which it may be comprised, should be able to adequately manage turns during its conversations with a user.

One of the most popular (and cited) methodologies for modelling turn-taking in conversation is the SSJ model named after its authors Sacks, Schegloff, and Jefferson (1974)—it is considered by some as influential and widely accepted (Kronlid, 2006). Whilst it is claimed to have shortcomings (Raux, 2008), the model provides a basis for understanding how turns are managed in human-human conversations. We present here a summary of the model as given by Kronlid (2006, pp. 82-83):

> A Turn Constructional Unit (TCU) is a phrase, clause, sentence or word with a predictable end […] [corresponding] more or less to an utterance. The first possible completion of a TCU constitutes a Transition Relevance Place (TRP)—a place where speaker-change is possible (or preferred). The turn transitions are governed by the following two rules:
>
> 1. For any turn, at the first TRP of the first TCU
>   a. The speaker may select the next speaker. In this case, the person selected is the only one with the right and obligation to speak.
>   b. Else, the next speaker may self-select. The first person to speak acquires the right to a turn.
>   c. Else, the current speaker may, but need not continue.
> 2. Rules 1 (a–c) apply for each next TRP of this TCU until transfer is effected

It is based on this model that Kronlid (2006) created a turn manager design for use in conversational agents, formalising the kinds of events that such a component would be expected to handle—a speaker starting or stopping, a speaker being expected to stop soon, or a speaker being addressed by another participant. Harel statecharts (Harel, 1987) are also used to specify the actions a turn manager takes with respect to the state of the conversation—who is speaking, and the state of TRPs. Statecharts are functionally identical to a finite-state approach to turn-taking, such as that given by Raux and Eskenazi (2012).

5.3 Action Selection

Once an adequate method of turn-taking has been established the system should now be able to structure the dialogue as a sequence of turns wherein each participant may get an opportunity to speak. One critical element remains: how should the system fill its own turns? This is the core problem of action selection which we detail in this section.

5.3.1 Definitions

In this section we refer to action selection as the process of choosing between possible dialogue acts, essentially deciding 'what to say next'. We distinguish between two kinds of action selection methods: handcrafted, and machine learnt. The ability of the former to decide what to say or do is based on decisions made during its conception by human developers or domain experts. Its decisions have been codified into rules or states, which are typically not modified or expanded upon during runtime. By contrast, machine learning (ML) approaches represent dialogue in terms of networks (Bayesian or Neural) or Markovian models and use techniques such as reinforcement learning, so that the dialogue manager is able to automatically learn strategies with datasets or during runtime with users. Lastly we recognise that some systems are hybrids which are combinations of separate approaches, but we do not cover them here as a topic, although we may give examples.

5.3.2 Handcrafted

Handcrafted systems utilise rules and decisions that have been programmed in majority or wholly by a domain expert or developer of the system. ELIZA, described earlier in §2, was a system that processed a user's utterance and produced responses that were the result of transformation rules, matched via keyword identification. Such rules were static and not added to, changed, or learnt during the system's interaction with the user; instead these were handcrafted. A frequently cited benefit of implementing handcrafted rules is the simplicity with which they can be produced. To clarify what is meant by simplicity, most systems of this type—such as finite-state machines discussed below—have a wealth of literature and are thus very well understood. The handcrafted rules or ontologies used to represent the domains are conceptually simple.

Because of this inherent simplicity, dialogue systems with handcrafted rules can be generated and applied in a relatively short time. Once the rules have been developed for the system, little additional processing is required during runtime as the system needs only to consult its dialogue strategies—from memory or from another data store—and use them in the dialogue.

Predictability is high in systems with predetermined rules and this may be an important aspect to consider in systems where such determinism is necessary. Critical domains that must be able to respond to each event in a precise way are clear cases for the use of finite-state machines. By using a system that operates only according to how it is specified, by experts, then it is likely to adequately respond to situations as per the conditions and constraints placed upon it—although formal verification of the system may still be required.

We will now present a review of some methods that have been used to implement dialogue managers with handcrafted action selection approaches.

5.3.2.1 Finite-state Machines

Finite-state machines (FSMs) are structures that have been studied rigorously in theory and in practice, including how they apply to spoken dialogue systems (Horacek & Wolska, 2005; Jurafsky & Martin, 2009; Lee et al., 2010; McTear, 2002; Sonntag, 2006; Stent et al., 1999). They can be summarised as a predefined sequence of steps that represent the state of the dialogue at any point during the conversation; each transition between states indicates pathways the system and user can take (McTear, 2002). FSM are archetypal handcrafted methods for dialogue management; they are developed solely through human developers and are rigid in the strictest sense, confined to highly structured tasks with inflexible dialogue flows (Lee et al., 2010).

Each state in the graph of a dialogue system's FSM represents actions that the system must take at a given stage; for example, it may ask the user to answer a question or decide what they wish to do next. Such systems can appear unnatural or at least non-representative of the dialogues that occur between humans.

Adding states to a FSM is itself a trivial process, but it becomes increasingly non-trivial when the dialogue system's domain is complex and wide-ranging. When cross-domain support and robust error correction is required, the graph of the system becomes intractable very quickly and the process of adding capabilities is rendered inefficient. It is a natural outcome that FSMs should be utilised in smaller domains that are not expected to grow and whose constraints and nature are defined very precisely.

Certain domains benefit substantially from using finite-states in their dialogue systems. Scenarios where errors must be handled consistently may benefit from dialogue paths which are deterministic. Predictability—although an attribute shared with other forms of handcrafted systems—is prominent in FSMs as any event can be traced back to the prior state which caused it. This characteristic may explain why McTear (2002) claims that most commercial systems utilise this form of dialogue control.

Realising the limitations that pure FSMs have in dialogues, authors such as Horacek and Wolska (2005, p. 1) utilise a creative hybrid between a FSM and an information state (§5.3.2.2) to create what they term the dialogue specification. Their domain, constructing mathematical proofs, appears naturally amenable to the finite-state approach and its adoption of the information state which they claim 'substantially [improves]' upon the rigidity of finite-states alone. Hybrids involving FSMs are common in the literature; an example is CommandTalk (Stent et al., 1999), a spoken-language interface to a military domain simulator, which utilises several technologies as part of its dialogue management capability including a plurality of finite-state machines in order to support varying kinds of dialogues.

5.3.2.2 Information State

The information state update (ISU) approach (Larsson & Traum, 2000; Traum et al., 1999), represents variables associated with dialogue state which are necessary to distinguish one state of dialogue from another. The approach is concerned with how these variables are changed over time through the use of rules which are applied at the discretion of the control strategy. The authors of the approach make clear the distinction between modelling a dialogue state intrinsically, such as might be achieved when moving through a finite-state graph, and the information state, which is kept explicit within the dialogue itself (a data structure separate from the dialogue model).

The information state approach has been used by many authors (Morbini et al., 2014; Mouromtsev et al., 2015) likely due in part to the accessibility of the TrindiKit (Larsson & Traum, 2000)—a toolkit for building and experimenting with dialogue move engines and information states—and the approach's natural amiability and cohesiveness with other dialogue methodologies. Earlier we made reference to CommandTalk, a system that combines this approach with the finite-state approach in order to lessen the negative implications of the latter and produce an overall improved dialogue system. Morbini et al. (2014, p. 151) have used the ISU approach with success in the practical system SimCoach to provide 'mixed initiative interaction' whilst also promoting 'efficient creation of dialogue policies by domain experts'. Given this description, it appears that the ISU approach does not always share the downfalls of handcrafted systems.

The use of a persistent information state that is kept across all sub-dialogues allows the application of rules and the production of dialogue acts based only upon the data that is kept in that state. The control strategy—the component that decides the process by which those rules and acts are selected—is core to deciding how the system reacts given the state of the dialogue at any given point. Given that the strategy is handcrafted, as are the rules, the ISU approach itself seems tenably deterministic and may suit domains where such determinism is of importance, whilst also achieving mixed-initiative dialogue.

The approach still necessitates the crafting of rules and strategies that must be able to handle any event of relevance to the domain. We mentioned earlier that in the system by Morbini et al. (2014, p. 151) the authors aimed to provide 'efficient policy creation' for their system, however they conclude that the process by which those policies were developed was cumbersome—requiring intense effort by multiple domain experts, and assistance by dialogue system experts. GALATEA, mentioned in Skantze (2007), maintains a state of dialogue similar to that of the ISU approach, but is claimed to be distinct in that its state is modular—the discourse modeller is reusable and domain-independent, whilst contextual information is separate. This degree of modularity may decrease the amount of effort required to move the dialogue system between separate domains whilst still maintaining the generalised dialogue model.

5.3.2.3 Rule-based

Rule-based approaches, as applied to spoken dialogue systems, are often compared with production systems (Webb, 2000)—a form of artificial intelligence composed primarily of IF…THEN rules used to allow reasoning by way of inference (Callan, 2003). The preconditions of such rules may be triggered by the context or state of the dialogue (Lee et al., 2006) or the user's input via pattern matching (Lison, 2015; Smith et al., 2011).

The system presented by Smith et al. (2011) uses a rule-based dialogue manager adapted from the work of Boye (2007) which captures processes in the domain as satisfaction rules.

These are satisfied if their sequence of sub-goals, actions, and conditions have been satisfied, executed, and proved true, respectively. These rules form the agenda, a tree-like-structure that the dialogue manager navigates through and expands in order to determine the next action to take. Among other reasons, the use of such a structure allows high-level rules to be executed as-is (without expansion) or with expansion to lower-level tasks.

OpenDial (Lison, 2015) is a hybrid—combining handcrafted rule-based dialogue management with statistically-based partially observable Markov decision process and Bayesian network optimisation models. It is the developer's aim that the system obtains the benefits from both kinds of approaches. The system's use of statistical rules has been claimed to be useful for three reasons according to Dragone (2015, p. 19):

- They are expressly designed for dialogue modelling. They combine the expressivity of both probabilistic inference and first order logic. This is an advantage in dialogue modelling where one has to describe objects that relate to each other in the dialogue domain and, at the same time, handle uncertain knowledge of the state variables.
- They can cope with the scarcity of training data of most dialogue domains by exploiting the internal structure of the dialogue models. By using logical formulae to encode the conditions for a possible outcome, it is possible to group the values of the variables into partitions, reducing the number of parameters needed to infer the outcome distribution and therefore the amount of data needed to learn the distribution.
- The state update is handled with probabilistic inference therefore they can operate under uncertain settings which is often needed in dialogue modelling where variables are best represented as belief states, continuously updated by observed evidence.

According to Webb (2000) rules are more flexible than script-based methods where dialogue must follow a fixed flow. However, they concede that rules are insufficient to model all kinds of dynamic dialogues performed by humans—and indeed their paper's focus is on interrogative and command dialogues. Given this, rules may have the most applicability in domains where the users are constrained to a predetermined set of acts, and not in cases where the user's speech is undirected such as in free conversation. Lison (2015) cites the ability for domain experts to express a system's dialogue domain in a 'compact' set of rules, which may increase readability for other system designers—thus serving as abstractions to the application's domain. Rules, although dependent upon implementation, can be specified generically such that they may be applied to any number of similar scenarios, thereby achieving a kind of abstraction of dialogue acts.

5.3.2.4 Frame-based

Authors such as Larsson (2002) and Jurafsky and Martin (2009) make no distinction between form-based and frame-based systems; both are synonymous with each other and describe dialogue where the role of the system is to encourage the user to provide answers to a set of slots, thus forming complete key-value pairs. McTear (2002) defines frame-based systems in much the same way: as those whose dialogue flow is determined only upon the user's utterances and the filled or empty status of remaining slots; they also use the term template-based, further confounding a unified terminology. For the purpose of discussion in this review we use the term frame-based for consistency.

A notable example of these systems is RavenClaw (Bohus & Rudnicky, 2003, 2009) developed at Carnegie Mellon University; confusingly it refers to itself as a plan-based dialogue management framework, seemingly in opposition to established definitions of plan-based systems (see §5.3.2.5). RavenClaw's dialogue models consists a number of dialogue task agents, arranged in a hierarchy; predefined agents exist for atomic dialogue acts (i.e. inform, request, expect, and domain operation) and an agent called an 'agency' for high-level structuring. Due to the nuanced approach the system adopts to dialogue flow, the frames need not be explored in exactly the way they were specified; instead, frames can be triggered earlier which results in a temporary jump of context. A full example of this is provided by the authors Bohus and Rudnicky (2009), explaining that this is achievable through 'concept binding', where information from the input can be bound to several 'slots'—not just the slot the system is currently asking for. This captures mixed-initiative dialogue flow as the user is free to provide more information than is required and at different times.

Frame-based systems, whilst still bound by a predefined set of information they must elicit, grant the user a degree of freedom as the questions the system asks need not be in a particular order (McTear, 2002) as explained in the RavenClaw example above. If there is flexibility in how the user provides answers to prompts, then some tasks may be completed much more quickly than usual; without having to ask or confirm more than is necessary.

It is claimed by McTear (2002) that frame-based systems lack the required expressivity to be used in domains whose tasks are ill-defined and where interactions with the user extend beyond the elicitation of predefined information (such as in negotiation scenarios). This view contrasts starkly with that of Rudnicky and Xu (1999) who instead see frame-based systems as offering a 'more flexible ' approach to the modelling of dialogue as compared to those which utilise fixed structures (e.g., trees). They justify by saying that the dialogue manager needs only monitor the frames; their completion, or lack thereof, can be used to dynamically engage in request dialogues for example.

5.3.2.5 Plan-based

The key underlying concept behind plan-based approaches is that each utterance (of the user or of another agent) should be treated as though it is an action performed in order to reach some goal (McTear, 2002), as congruent with the research of Perrault et al. (1978) and Grosz and Sidner (1986). Without going to great depths with plan-based theories of speech and discourse—for which we direct the reader to other research (Chu-Carroll & Carberry, 1994; Moore & Paris, 1993)—we cover systems that make claim to the implementation of a plan-based approach.

By identifying the overall goal the user wishes to achieve, the system can develop a plan—composed of a series of (dialogue) actions—that it believes will link the current state of the conversation to the achievement of their goal (Skantze, 2007). Plan-based modelling of dialogue thus involves breaking down the overall task into smaller goals and plans, and controlling the interaction to accomplish them and therefore the overall task (Wu et al., 2001).

RavenClaw and Topic Forest, the latter to be mentioned in the next paragraph, both match users' utterances or the system state to a particular frame in a tree, and this is used to infer the user's goals. This kind of implicit goal reasoning is not excluded by the definitions of the plan-based approach, but at the same time does not appear faithful to the formal theory. A criticism that may be made of these two particular systems is that they do not actively develop and expand upon plans during runtime—a feature typically associated with plan-based approaches—and are instead already present within the system. We take the view that there may be an overlap: frame-based systems can be one way of achieving plan-based dialogue; frames structured in a hierarchy, for example, are naturally amenable to plans which can be de-constructed into lower-level goals—fitting the previous definitions.

Topic Forest (Wu et al., 2001) is a plan-based dialogue management structure that utilises hierarchical relationships (topic trees) to represent the information items required of different domain topics. Aside from the domain-dependent tree structure, it features a domain-independent reasoning engine and strategy (i.e. action selection) that consults the structure of the topic forest for nodal information.

We also believe the conversational agent system described by Smith et al. (2011, p. 9) exhibits some degree of plan-based behaviour, given that its components use a Hierarchical Task Network which 'works through recursive decomposition of high level tasks into sub-tasks'—a behaviour that appears to follow the definition of plan-based approaches presented here.

Plan-based methods have the ability to provide scalable solutions to dialogue management, containing the required intelligence to automatically decide the pathways through a conversation (Wang, 2000).

5.3.2.6 Agent-based

Agent-based systems view dialogue as an 'interaction between two agents, each of which is capable of reasoning about its own actions and beliefs, and sometimes also about the actions and beliefs of the other agent' (McTear, 2002, p. 94). With this method, the SDS is rarely referred to as a single 'system', consisting at least one software agent that communicates with the user; this agent is designed with characteristics that would be present in a human partner. A rationale for the use of agent-based systems has been the recognition that certain problem-solving tasks involve a cooperative effort between individuals—this is supported further when agents have differing capabilities. The embodiment of agents, including their reasoning and 'intelligence', is discussed in greater detail in §5.4.

There is no 'agent-based' method of action selection, as the term actually refers to software architectures that stipulate dialogue (on the part of the system) as being composed of smaller agents which must contribute toward a unified response. Importantly, the method of action selection within a single agent may actually be different to that used within another agent—such as frame-based in one and plan-based in another. An action or response in an agent-based SDS is the outcome of the combined contributions of each relevant[12] agent which have engaged in a collaborative activity based upon the rules of engagement and cooperation that have been instilled within them.

Lin et al. (1999) describes a proposed multi-agent architecture for a dialogue system whose key aims are to ensure domain extensibility and maintain a distributed agent construction. Here, agents are organised in a 'society' that is distributed and interconnected, with a user interface agent operating between the agents and the user. Multi-agent systems are naturally amenable to multi- and cross-domain systems as implementation can be designed such that each agent is an expert in a particular segment of the domain, as was the aim of Lin et al. (1999) above.

We conclude our review of handcrafted approaches and their applications to dialogue management in spoken dialogue systems, and now move to discuss systems whose action selection methods are based upon machine learning algorithms.

5.3.3 Machine Learning

In this section we describe the methods in which machine learning (ML) techniques are often applied, although generally they may be termed data-driven (Lee et al., 2010) due to their use of large datasets in order to learn dialogue strategies. The use of ML and statistics in this context is relatively new, pioneered by many authors (Lee et al., 2010; Lemon, 2011; Lison, 2015; Williams & Young, 2007; Young et al., 2007); a full listing of such authors is not given in this précis of the topic. These systems may be considered dynamic as often they have the ability to apply their learning algorithms whilst interacting with the user, although they require some form of bootstrap process (e.g., reinforced learning) before they can communicate usefully.

Different kinds of ML methodologies exist for dialogue management systems and they differ significantly in how learning occurs, although generally the output of each is always a recommendation for which dialogue action to take next. Neural networks, for instance, are composed of a number of nodes which may be arranged in any configuration.

ML techniques have gained popularity due to their ability to automatically decide what dialogue act to choose at any point, based on prior learning (i.e., during the bootstrap process or prior conversation), with the aim to reduce the necessity of domain experts to constantly add to the dialogue strategies and rules. Successful implementations of such systems take a corpus of input data (e.g., conversational data relevant to the domain) such that the system can learn appropriate responses to certain input utterances from the user. The system may learn with real users during runtime—with 'feedback' interpreted from the user's responses—thus achieving adaptation and personalisation to the user or a group of users. Some ML systems may claim to be extensible due to the learning processes being domain-independent; however, the corpora from which they bootstrap may not be.

[12] Whether an agent is 'relevant' for input to the response may be determined by its domain knowledge. For instance, an agent specialised in a maritime domain may not be allowed to contribute to answering a query about an event happening on land. Other implementation-specific measures of relevancy may also be used in addition to the knowledge of agents.

One of the primary disadvantages of these systems is their reliance upon data to support the learning processes; if the system is not provided with substantial datasets, then its action selection decisions will be inaccurate and result in incorrect responses. System development complexity tends to increase in ML systems as they must be programmed and configured to conduct similarly complex inferencing and calculations with significant mathematical overhead. Lastly, it becomes extremely difficult to predict the output of a dialogue system that utilises ML algorithms as, due to their nature of making background calculations, it is impractical to observe what values are being used and thus how a system came to a particular conclusion (e.g., the choice of a certain dialogue act).

We now explain different ML mechanisms that have been employed for the selection of dialogue actions.

5.3.3.1 Bayesian Networks

Bayesian networks (BNs) are probabilistic structures that capture probabilistic distributions between events or variables, and comprise two parts: a directed acyclic graph, and conditional probability tables for each node (Lee et al., 2001). Bayesian networks are generally applied due to the realisation that the environment in which SDSs operate is inherently 'noisy'– the users' utterances can be unclear due to unnecessary prolixity or speech recognition errors. Situations where Bayesian inference can assist include keyword and feature recognition (Wollmer et al., 2010), and in user modelling and intent recognition (Hong & Cho, 2003; Horvitz et al., 1998; Lee et al., 2001). The use of BNs for deciding system actions appears to occur only when it is combined with other methods, in particular Partially Observable Markov Decision Processes (POMDPs) (Jurčíček et al., 2011); in such systems the BN does not decide the action, though it provides useful information such as a model of the user's intents. Even a system such as that described by Chien and Chueh (2012), which uses a 'variational Bayes' procedure to segment natural language into topics, is combined with Markov chains.

Research by Lee et al. (2001) has proposed the use of a Bayesian network to assist with intent recognition for a plan-based SDS. A form of user model is given that creates causal relationships between words in a user's utterance, and the likely goal the user has (if they used those words). This SDS contains a goal inference module whose purpose is to deduce the goals of the user via Bayesian inference; if it was unable to, for instance due to lacking information, then it could instruct the dialogue manager to issue a correction action to the user. A later paper was produced, detailing a similar system (ostensibly referred to as 'conversational agent') (Hong & Cho, 2003).

In a similar way to handcrafted systems, the network structures in BNs are designed by domain experts or developers and so face many of the issues the former have: time and effort of development, and domain inextensibility (Hong & Cho, 2003; Lee et al., 2001). The nodal structure of the BN must be specified by a human developer, and the initial conditional probabilities must also be calculated. These tasks must be done during the system's inception but also whenever its capabilities must be extended, including if the domain must be changed entirely.

These establishment overheads can be reduced if the domain in which the network is applied is suitably small and manageable—the BN approach is not trying to capture the entire dialogue model—and if it is combined with other techniques, notably Markovian models. If the domain was indeed small enough, then one could posit that a FSM could also be applied; however, if the transitions needed to be updated in response to observed interactions or other training data, then a BN would still be necessary.

We recognise that a desire made clear by the literature is for further research to be done in the area of automatically generating the Bayesian network structures, without human intervention (or very little), and for probabilities to be created in a similar manner. Efforts toward this end have not been evaluated and are not discussed here. Some, such as (Lim et al., 2010, p. 92), have claimed there exists a deficiency in the ability of BNs to deal with truly dynamic input, claiming that their 'predefined methodology' is restrictive and unable to 'change topics naturally'.

5.3.3.2 Neural Networks

In the context of spoken dialogue systems, neural network (NN) approaches tend to feature less in dialogue management but are especially prominent in speech recognition and natural language processing areas for processes such as sequence matching (Hu et al., 2014), learning (Meng et al., 2015), and prediction (Mingxuan et al., 2015). The literature survey conducted for this report uncovered very little research in the area of dialogue management with specific use of NNs for the purpose of action selection, which the authors felt surprising—given that NNs have shown increased popularity in recent years.

NN techniques have frequently been applied to output generation via corpus learning (Shang et al., 2015; Sordoni et al., 2015) within the domain of microblogging sites (i.e. Twitter, Weibo) where the data sources are abundant and the expectations are the production of short-text conversations. Despite positive results in such implementations they appear restricted to single-round responses, albeit as their design intend, and appear unsuitable where conversational interaction is needed.

Although not strictly describing a NN, Lim et al. (2010) present a system capable of providing flexible mixed-initiative interaction with the use of semantic networks, a global workspace theory, and representations of memory. Here the focus is on being able to dynamically switch between topics dictated by the operation of the semantic networks and a 'spreading activation process'. We believe that this application, whilst it does not use a true NN, hints to the possibilities that NNs may have for action selection.

We believe the use of NNs for action selection should be a future goal for research—as common applications of the technique appear grounded in natural language processing. In fact, NNs may form one part of the system that assists in the bootstrapping process, by capturing models which can later be used to train Partially Observable Markov Decision Process models—reducing the need to develop a task-specific dialogue corpus (Serban et al., 2016).

5.3.3.3 Markovian Models

In this document we use the term Markovian model to refer to any of the following: Markov Chain (MC), Hidden Markov Model (HMM), Markov Decision Process (MDP), and Partially Observable Markov Decision Process (POMDP).

Markovian models have been applied many times in SDSs and DMs for their 'principled mathematical framework' in modelling levels of uncertainty within spoken dialogue systems (Williams & Young, 2007; Young et al., 2007). POMDPs used in dialogue management may also positively affect robustness in terms of automatic speech recognition and natural language understanding (Lee et al., 2010). A fundamental concept with Markovian models is the 'formalisation of dialogue as an optimisation problem' (Jurčíček et al., 2011; Levin et al., 2000)—where a system must choose the *optimal* action at any given point in the dialogue. The metrics used to decide the optimal action are usually the costs incurred by the system if it selects a particular action, but these costs differ between authors. Levin et al. (2000) suggest that dialogue duration, resource access or use times, user satisfaction, and others. Williams and Young (2007) in their use of POMDPs implicitly capture what system actions are desired by associating them with large positive rewards, and negative rewards to ill-favoured strategies (e.g., deleting information when the user wanted to save). Costs and rewards are purposefully ambiguous as they are defined by the system developer.

Markov model-based systems require some form of training in order to learn dialogue strategies. Supervised learning is one means to achieve this, which attempts to estimate a direct mapping from a machine state (which may just be a recognised user input) to a system action by utilising a corpus of 'training examples' (Williams & Young, 2007). In the case of Williams and Young (2007), the authors utilised a corpus of human-human dialogues in natural language where the interactions were altered to simulate speech recognition errors. Once a basic strategy has been developed, the Markov model can still be updated using the same ML techniques but in this case data will be sourced from the user—system states will now be utterances of a real user, whilst feedback for the system's actions may be elicited through explicit or implicit means.

Papangelis et al. (2012b) take Carnegie Mellon University's Olympus system and extends its dialogue management module (RavenClaw) to utilise online[13] reinforcement learning algorithms—which use MDPs as a model—in order to learn optimal dialogue strategies. Their particular use of these algorithms, its developers claim, awards benefits such as: simplicity of implementation, low computational cost, and the ability to optionally use handcrafted rules. They present an example dialogue with the system, before and after training with a particular reinforcement learning method, which shows clear improvements—especially in terms of efficiency and repetition.

Despite the volume of research successfully using Markovian models and reinforcement algorithms in calculating best dialogue strategies, there are a number of issues to be considered. By adopting strategies that have been created automatically by the system, without a human-in-the-loop developer, the system has essentially removed control from the developers to ensure that dialogue flow is effective and suitably refined (Lee et al., 2010). In the system by Papangelis et al. (2012b), the system allows for the domain expert to create and apply handcrafted rules which grants them a greater ability to ensure the conversation is adequately constrained.

[13] Online algorithms can be applied during runtime as the dialogue progresses, as opposed to offline algorithms which calculate an optimal policy and use that during interaction with the user (thus it is static).

A major criticism of data-driven techniques is their reliance upon a substantial corpus (or corpora) to train the system effectively in a particular domain if a suitable corpus of conversational data is not available, then the system is not viable, and alternatives must be sought such as Wizard-of-Oz experiments[14] which are also costly.

5.3.4 Summary of Strategies

For a spoken dialogue system to appear coherent and to be effective in its tasks, it must be able to choose what action to take given what the user has said and the context of the dialogue as a whole. Naturally, the years of research in this area has led to a plethora of methods by which a dialogue manager can choose its next action. We have broadly classified these into two groups: handcrafted and machine learnt; the former typified largely by the efforts of domain experts in explicitly specifying the DM's actions, whilst the latter by ML algorithms and associated modelling techniques in making such decisions automatic.

As shown in this section, many handcrafted methodologies have been used to create dialogue systems. We also acknowledge that there are other systems whose approaches to action selection are the result of a hybrid between different methods. Of particular interest is the spectrum of capability within these systems, ranging from completely static and unchanging finite-state-based models of dialogue to adaptive and mixed-initiative-capable methods such as frame- and plan-based systems. We observed that while the action selection strategies of handcrafted systems require experts, the systems are still able to display behaviour that appears dynamic and robust.

Benefits of handcrafted systems will vary depending on the requirements of their domain of implementation and the capabilities desired. Systems concerned with security, safety, or strict adherence to business rules necessarily require the ability to adequately predict and cater for expected and unexpected usage scenarios; here, the key characteristic is determinism. In addition, handcrafted methods are easier to implement in smaller domains and simpler use cases, and their outputs can always be derived back to the conditions and inputs that caused them.

Other methods have favoured data-driven techniques of dialogue management. We have termed them ML systems as they typically involve some form of ML technique applied to a large dataset, such as a corpus, in order to learn correct responses. Rather than relying solely upon the efforts of domain experts and system designers in producing rules or graphs, they derive their conversational abilities from corpora and reinforcement or supervised learning algorithms.

A defining characteristic of these systems is their reliance upon large datasets (e.g., corpora of natural language interactions between humans) in order to produce sufficiently reliable dialogue strategies, and this can be positive or negative depending upon domain of implementation. Availability of corpora and training data may be plentiful in social

[14] A Wizard-Of-Oz experiment is an interaction where a participant acts in the role of a user, and a researcher acts in the role of the system. It is through such an experiment that example human-computer interaction data is collected.

media scenarios[15] (Serban et al., 2016; Shang et al., 2015; Sordoni et al., 2015), but this may not be the case in specific domains; without appropriate resources the ML algorithms cannot operate effectively. With appropriate training, however, they are able to respond to inputs in ways that cannot be matched or anticipated with rules prior to deployment. Indeed, they are perhaps sought after for their ability to operate without extensive effort by designers, and to adapt with extended use.

Hybrids of different strategies may be utilised in order to capitalise upon the benefits of more than one approach; for instance, it is tenable that an FSM could be paired with a form of ML in order to tweak the transitions between states as the system interacts with the user. Although the 'model' of the dialogue may have been hardcoded initially, it is adapted automatically throughout interaction.

5.4 Multi-agent Architectures

In §5.3.2.6 we discussed that some dialogue managers may implement multiple agents that work together in order to facilitate complex interactions with users. In this section we continue the discussion by characterising the agents in such architectures, including: how they represent their knowledge, perform reasoning, and how multiple agents negotiate and cooperate on assigned tasks.

In order to ground our discussion of agents, we must first define precisely what is meant by the term agent. Although several definitions exist, Callan (2003) takes the view that they are goal-oriented entities capable of autonomous action within a certain environment, and, in order to be able to achieve their goals, they must also be able to perceive and respond to that environment. Russell and Norvig (2010) define agents as anything that can be viewed as perceiving its environment through sensors, and acting upon that environment through actuators. Jennings et al. (1998) consider three key concepts relevant to defining agents: situatedness—being responsive to its environment, autonomy—acting of its own volition without external intervention, and flexibility—being responsive, pro-active, and social.

Multi-agent systems can be thought of as systems with: 'interacting autonomous agents that when acting together have more capability than any single agent within the system' (Callan, 2003). In the literature there are many reasons put forward as to why multiple agent architectures are beneficial. Some highlight their ability to adequately model complex systems (Callan, 2003), and others (Jennings et al., 1998) describe the increased utility when multiple agents learn in a system.

We begin by considering a common agent architecture whereby the agents are programmed with internal notions such as beliefs, desires, and intentions. The set of actions (including dialogue acts) that is accessible to any agent at any time is governed by these notions. Further, we present the general principles that have been established as

[15] Social media domains typically contain short-text utterances only, and are limited from the perspective of extended discourse, so may not be appropriate for dialogue management.

necessary for sensible and cooperative multi-agent systems, and provide several theories that have been proposed to achieve them in practice.

5.4.1 Agent Intelligence

In order for agents to make sense of their environment they must have some form of knowledge of that environment, and a corresponding reasoning ability over that representation; for the purposes of discussion, we here refer to this as an agent's *intelligence*. A common trend in this area has been the development of human-like agents whose representations of the world are similar to those of humans. A popular mechanism for the representation of agent knowledge and reasoning is the beliefs, desires, and intentions (BDI) model initially defined by Bratman et al. (1988). In this reasoning architecture, agents are characterised as having the following mental states:

- Beliefs: what the agent believes to be true or untrue about the world (including the environment, its own existence, and that of other agents)
- Desires: roughly translated as the agent's goals; what changes or end states does it desire of the world; Jennings et al. (1998) interpret desires as an agent's options—what it could commit to
- Intentions: once an agent decides to bring about a desire to fruition, it is essentially intending to commit an act (or several) to ensure that it is realised.

In this model, an agent is considered to exist in an ongoing loop of: perceiving the world, identifying tasks to perform, decomposing them into concrete actions, and performing those actions to completion. Throughout, the agent will adopt desires to complete tasks, and therefore have intentions to undertake certain actions, and will have its beliefs about its progress and the state of affairs in the world. The model is common in the field, with variants being adopted in other systems.

Bretier and Sadek (1996, p. 202) specify a multi-agent system whereby the behaviour of the agent, its ability to communicate and cooperate with others, is derived from the 'normal reasoning processes based on generic rationality and cooperating principles'. An agent's mental model is based on a formal theory of interaction written in a first-order modal logic of attitudes—belief, uncertainty and choice. The model is applied in the *ARTIMIS* system. Axioms within the theory explicitly define what constitutes rational behaviour, communication, and cooperation; this system consists of an inference engine (theorem prover) that utilises these axioms and other rules of the theory to achieve reasoning.

Schubert (2005) makes the case for explicit self-aware agents based on characteristics of humans. He argues such an agent should have knowledge of: itself; its history, environment, goals and intentions; and other agents. The term 'explicit self-awareness' is made clearer by the requirement that an agent's self-awareness should be transparent to others and able to be communicated (in natural language). It is argued that the best representation for such explicit self-awareness is episodic logic, which is known for its natural language-like expressiveness (Schubert, 2005).

It is critical to specify rational sets of behaviours and axioms for agents to ensure that they conform to expected behaviours during interaction. Without a specified theory of reasoning, the effectiveness of the spoken dialogue system as a whole is reduced. Human-like reasoning is important for agents in an SDS that interacts with users.

5.4.2 Cooperating and Negotiating Agents

In order for multiple agents to cooperate and/or negotiate so as to respond to a user's queries, there must be theories dictating how that cooperation and negotiation takes place. We present the general principles that are necessary for the development of sensible cooperating and negotiating multi-agent systems, and provide several theories that have been proposed to achieve them in practice.

Cooperating or negotiating agents need to communicate with one another. The most influential formalism for dealing with communication between agents is speech act theory—popularised by Searle (1969)—where communications are treated as actions, whose properties are characterised by pre- and post-conditions. Agents can negotiate to come to agreement on matters of interest, or they can cooperate to solve problems. Negotiation usually proceeds in a series of rounds, with each agent making a proposal at every round—the proposals that agents make are defined by their strategy. They are drawn from a 'negotiation set' and are often required to be legal—conforming to agreed rules of interaction. If an agreement is reached, then the negotiation terminates with an agreement deal struck.

As Jennings et al. (1998) highlight, the process for planning the actions of a single agent requires only consideration of that particular agent's internals (e.g., its beliefs, desire, etc.) and its own environment: this is not the case for multiple agents attempting to negotiate. Such a scenario is faced with numerous complications, some of which include the negotiation over multiple attributes, and the plurality of interaction—whether one-to-one, one-to-many or many-to-many (Wooldridge, 2009). Also, an agent may be required to justify its position, or even change its position.

Cooperation between agents when problem-solving usually involves three stages. In the first, the problem is decomposed into smaller sub-problems; this can be done by the group as a whole or may be conducted by a single agent. In the model presented by Ferguson and Allen (2011) a shared decomposition can occur wherein the first agent (the one to receive the task initially) can suggest to a second agent that the task indeed be shared. The second agent may respond positively or negatively to this proposition—the latter case may result in the first agent choosing to complete the task itself, or proposition a third agent. If the proposal is successful, then further propositions will occur to decompose the task to each agent's approval, at which point each sub-task (with responsibility delegated to either the first or second agent) can now be completed; this is the second stage of problem-solving. The final stage may involve a final 'submission' where the combined results of each sub-task are combined to produce a solution to the original higher-level task; this may be given to a separate entity, such as a task manager.

5.5 Understanding the User

A dialogue system's raison d'être is the ability to accept spoken input from the user and correctly acting upon their queries or commands. In order to do this however, a determination must be made about what exactly the user intended. Relying on speech is not always sufficient to infer the user's goals, and so additional information is required to act with some higher degree of certainty. In this section we highlight two key considerations for a DM: user modelling and intent or goal recognition.

5.5.1 Intent & Goal Recognition

5.5.1.1 Definitions

SDSs are the interfaces to functionalities the user is interested in interacting with, but for SDSs to help users reach their goals, they must have a basic understanding of what each user desires. Both 'intent' and 'goal' are used to describe these desires, so here we present a brief definition of key terms as necessary to understand the relevance these themes have for spoken dialogue systems.

Before discussing alternative perspectives, we first acknowledge the seminal works of Grosz and Sidner (1986) in which they define intentions as a general terminology to refer to the purpose of a discourse. In their discussion, intention is used in the definition of two key concepts: the discourse purpose and discourse segment purposes; both critical to the intentional structure component of their discourse structure theory. We summarise Grosz and Sidner (1986, p. 178) and their treatment of intent by extracting from their definition of discourse purposes: 'intention provides both the reason a discourse (a linguistic act), rather than some other action, is being performed and the reason the particular content of this discourse is being conveyed rather than some other information.'

We have observed in the literature a complementary discussion of both intent and goals, often with the former specified in terms of the latter. For instance, some such as Higashinaka et al. (2006, p. 2) would see user intention defined as 'the information that the user has in mind to convey to the system in order to achieve [their] goal'. This definition claims that intents and goals are separate entities altogether from which one can deduce that goals refer to the overall purpose for the user being engaged with the system. This is also reinforced by Bhargava et al. (2013, p. 1) who specifically define intents as 'global properties of utterances' that 'signify the goal of the user'; additionally noting that the goal may change across domains and over time. Lastly, Bratman et al. (1988) claims that intentions are active determinations to achieve goals and are the result of a deliberation process whose inputs are an agent's beliefs and desires.

Given the above, there appears to be a clear distinction made between intents and goals; in fact, the former appears to be defined in terms of the latter. Unfortunately this is not always the case, as Perrault et al. (1978) define 'goal' independently of intention—an overall end sate that an agent wants to achieve. Kass and Finin (1988) also describe this end state as a 'state of affairs'. For the sake of leaving semantics to the experts, we do not redefine any terms here and consider both 'intent' and 'goal' to be synonymous in our treatment of spoken dialogue systems and dialogue management—both referring to the desire of a user to achieve something, thus providing the rationale for their use of the SDS.

5.5.1.2 Application

The focus for all applications and usages of intent in spoken dialogue systems is in understanding the user. Callejas et al. (2011), for instance, detail such a system which includes a mental state prediction unit that acquires intent and emotion in order to make an estimate of the user's mental state at the current point in the dialogue.

As previously noted, being able to discern what the user is trying to do is of utmost priority to the spoken dialogue system; this is especially true for the dialogue management component which may be able to utilise that information to adapt the conversation to those intents (Callejas et al., 2011). More simply, tracking the user intent may be seen as a way of identifying what the user wants given a particular utterance and delivering that function. Bhargava et al. (2013) explains this notion with a software development analogy: comparing user intents as functions—which may be executed by the system—and the goal is to correctly identify which function the user wants the system to execute.

5.5.2 User Modelling

5.5.2.1.1 Terminology

It is Fischer's claim that the objective of human-computer interaction (HCI) research is to 'make systems more usable, more useful, and to provide users with experiences fitting their specific background knowledge and objectives'. User modelling is thus a means by which these goals can be achieved, and this has an impact on dialogue systems—due to SDSs being a form of HCI.

According to Zukerman and Litman (2001), the origins of *u*ser modelling for dialogue systems can be traced to the works of Kass and Finin (1988) and Wahlster and Kobsa (1989). Kobsa (2001) later claimed the concept was the result of first efforts by Perrault et al. (1978), and, separately, Rich (1979). Given the number of authors in this area, all of whom are credited with founding or refining the concept, it becomes an arduous task to settle upon a universal definition that captures what user modelling is. 'User Modelling Inc.'[16] is a society for researchers and practitioners for user modelling, and it defines the term as 'any aspect of systems that acquire information about a user (or group of users) so as to be able to adapt their behaviour to that user or group.'

We begin with Kass and Finin (1988) who state a tentative definition: 'a user model is the knowledge about the user, either explicitly or implicitly encoded, that is used by the system to improve the interaction'. They argue that such a definition is strong in that it implies only the user of the system is modelled, and weak due to its consequence that many systems could be considered to have user models even if they are implicit in nature. They distinguish between agent models, which model individual entities regardless of whether they interact with the system, and the sub-class user models which model users interacting with the system.

[16] http://www.um.org/. User Modeling Inc. Accessed 15 November 2016.

Perrault et al. (1978) take the view that each agent has a model of their internal beliefs and knowledge, but also that of each other agent of which they are aware. Kass and Finin (1988) make no distinction between a user (even human) and an agent; they are both expected to have a model of one another. Both papers introduce the idea that modelling should be broad, generic and applicable to any kind of agent or interlocutor.

In contrast to the modelling of agents given by the authors above, Rich (1979) elaborates upon stereotypes which contain facets (characteristics) relevant to the domain of implementation and whose values are typical of a particular class of user. This characterisation of a user model is much narrower as it models only what the user *is* and not what the user *knows* or *believes.* Wahlster and Kobsa (1989) state this kind of representation lacks the expressive power required for elaborate dialogue systems. The theory of agent modelling given by Perrault et al. (1978) does provide such expression through the complex representation of an agent's knowledge of the world including of itself and others.

We believe the notion of agent models proposed by Perrault et al. (1978) is broader than the stereotypes considered by Rich (1979) which would exist as subsets of a particular agent's knowledge. Wahlster and Kobsa (1989) emphasise a similar distinction between simply containing data about a person and actually recognising that data as describing assumptions of the user by the system. Thus, we take the view that a model of an *agent* (user or otherwise) may consist of its own knowledge of the world, its understanding of others' knowledge of the world, in addition to the facets described by Rich.

5.5.2.2 What's in a Model?

What is included within a user model is typically application-specific and depends on its requirements (Wahlster & Kobsa, 1989); there is no clear blueprint from which they are developed. Kass and Finin (1988) also come to a similar conclusion. Having observed the different varieties of user models in existence, they summarise them as not being a 'homogenous lot'. Several authors, as discussed below, have proposed a number of dimensions that may be applied to the models.

In their review Wahlster and Kobsa (1989) summarise two systems, GRUNDY (Rich, 1979) and UMFE, which hold assumptions about the user as 'linear parameters' which may be summarised as a set of key-value pairs (with an optional 'level of certainty'). These assumptions are characteristics of the user such as knowledge of a particular domain concept, or 'attitudes'. More recently, a system by Papangelis et al. (2014) models the user's emotional state within a 'user profile system' along with other information relevant to its domain. The rationale for a user model in the latter system is to give users the appearance that it was 'intelligent', and it was hoped that remembering details of the user (across multiple dialogue sessions) will contribute toward this end. These systems utilise one-dimensional representations of the user's characteristics and thus embody the narrow interpretation of user modelling as previously discussed.

Kass and Finin (1988) distinguish four categories of knowledge that can be contained within a user model: goals and plans, capabilities, attitudes, and knowledge or beliefs. Goals and plans (often synonymous with intents) are frequently cited as being of key importance in any spoken dialogue system, as this mirrors the same actions present in regular human-human conversations. By recognising what the user wants, in immediate or eventual terms, the system can match that to one of many patterns or series of intermediate goals that it can apply to reach those goals (and indeed those of other agents or itself). Similarly, the capabilities of the user can also affect how the system speaks to them. In considering the kind of response it gives to the user, it must also consider whether the format would be appropriate and understandable by them—syntactically, lexically, or semantically. This is the same as considering the user's attitude which also has the effect of impacting the system's choice of response, though in that case it is due to the user's 'preferences' or personality. Finally, Kass and Finin (1988) elaborate the kinds of knowledge the user has: of the domain, of the world, and of the other agents. These are assumptions about what the system believes the user to believe and can be applied to correct misconceptions by tailoring a dialogue to the user's belief set or by drawing on useful worldly reference points.

5.5.2.3 *Issues with Modelling*

Whilst user models have been designed and employed as a means to achieve improved user-adaptive dialogues, like any other major software component they too experience a number of design and implementation concerns that can affect the return expected of them.

The inclusion and use of a user model naturally implies the desire for the system to adapt to the needs of the user or to utilise a representation of their knowledge. The logical consequence of this is the tendency of the spoken dialogue system to become non-deterministic, as stated by Johansson (2002, p. 19) who claims that this may have a negative impact on the usability of the system as users feel 'without a sense of control'.

User models can contain incorrect information, and it is important for the system to be able to identify when a model is inadequately reflective of the users or environment and then act using whichever modification mechanisms are available Fischer (2001). An important question is put forth by Kass and Finin (1988, p. 17) who ask, 'how will an error in the user model influence the performance of the application system?' This illustrates how, under certain circumstances, it may not always be appropriate to make decisions based upon the model. ML and statistical models have been proposed as solutions to this problem and appear promising according to Zukerman and Litman (2001) in their application to the creation and update of the models.

Kass and Finin (1988) also make a critically important observation of the relationship between the complexity of the implementation domain and subsequently the complexity of the user model. If the purpose of the model is to identify the user's plans, for example, then a simple strategy might be to conduct a search operation on a repository of all plan permutations; this becomes intractable. A more reasonable, and computationally feasible, alternative is instead to consider the likely plans and conduct inferencing upon those—thus reducing the large processing overheads.

Early on in the design of the user model, the granularity of representation must be established in accordance with the expected variety of users interacting with the system. Kass and Finin (1988) suggest that if users do form a 'homogeneous group', then a generic model may be adequate, and this need not even be explicit. On the other hand, the situation where unique users have radically different 'characteristics' may be handled by an appropriate explicit user model as defined in this section. A generic user model should be considered distinct from Rich's use of stereotypes (Rich, 1979) as it instead refers to a complete model of the shared knowledge, beliefs, attitudes, and capabilities of an entire group of users.

Lastly, we recognise that important and interesting social, moral, and ethical issues have been discussed in the literature but these will not be elaborated in this review as they fall outside of the scope of the current review; instead see (Johansson, 2002; Kobsa, 1993, 2001; Wahlster & Kobsa, 1989) for discussion on these topics.

5.6 Error Handling

5.6.1 Why Handle Errors?

Before a spoken dialogue system can respond to a user's requests, it must be able to understand what they are saying, but, as McTear (2002) highlights, actually determining what the user wants can be problematic for a number of reasons. The input— typically text or speech— may arrive at the system incomplete or inaccurate, thus it may not have enough information to be sufficiently interpreted to act upon. Generally, errors in speech systems can be attributed to the inability to recognise the user's utterances during speech recognition or incorrectly identify their meanings in language understanding, due to reasons such as lack of information or ambiguity Lee et al. (2010). Without an internal understanding of the input that is congruent with what the user actually said, any processing or action undertaken with that incorrect representation will consequently lead to an inappropriate response (Lee et al., 2010).

Furthermore the presence of errors (especially their frequency) is often used as a means to evaluate the effectiveness of spoken dialogue systems. Metrics in such evaluations include: ASR word error rates, number of confirmation utterances, number of exchanges between a user and a system (Griol et al., 2016), turn length, dialogue success rate (as opposed to failure, indicative of errors), and others (Lee et al., 2010). For an SDS to be considered effective and useful in its domain it must have a reasonable error correction capability to survive the scrutiny of evaluation, as well as the uncertainty of interaction. For the purposes of this review, the methods and findings of SDS evaluation literature will not be covered here; see (Lee et al., 2010; McTear, 2002; Walker et al., 1997) for additional treatment of that topic.

5.6.2 Types of Errors

The most documented errors in dialogue systems are those which are the result of the automatic speech recognition (ASR) devices; they are sometimes called errors of mishearing (Jurafsky & Martin, 2009). Skantze (2007) uses the broader term 'miscommunication' with two outcomes: non-understanding, the result of the ASR being unable to select a single hypothesis for what the user has said; and misunderstanding where a hypothesis has been chosen but it does not fit what the speaker intended. The former often means a complete and utter failure in the system's ability to understand the words and thus the meaning of the user's statement; the latter indicates failure in just meaning, but this may not become apparent until later in the dialogue when the system begins to make assumptions based on the previous misunderstanding.

Once a theory of the user's utterance has been established—whether correct or not—it must now be parsed with downstream language processing techniques; Jurafsky and Martin (2009) summarise this process by listing the following knowledge required of the system for this stage: phonetics and phonology, morphology, syntax, semantics, pragmatics, and discourse. Natural language input from the user can be erroneous in many of these areas. Ambiguity for instance is a problem in spoken dialogue systems—often requiring the use of context to be resolved—including deixis, anaphora, and illocutionary acts. These ambiguities are not apparent during speech recognition but will become so during the process of natural language understanding—when the system must make a decision about what the utterance meant in terms of semantics and pragmatics.

Language issues can also become the responsibility of the dialogue manager—although it may deal with low-level errors as well. The DM may call upon its various knowledge sources (dialogue history, and various domain and user models) to resolve ambiguity not previously handled adequately by the NLU component and also may place utterances within a relevant context. The DM is in the unique position to understand the user with greater accuracy as it may utilise information gathered across multiple utterances, topics, and even conversations, thus improving its confidence in certain hypotheses.

The ability to maintain the common ground is a fundamental concept for spoken dialogue systems as all contributions to the conversation contribute to the shared ground between two or more participants. The common ground also represents the extent of understanding any one participant has, and upon which all other correspondence must be based in order to understand future dialogue acts.

Optimally natural language understanding modules must handle errors which can be attributed to malformed input in any of these areas. Failure at earlier stages in the process can propagate forward into the dialogue manager, and even out to the user if the output was based on misunderstandings; however fatal errors in language can prevent further processing altogether and grind the system to a halt. All errors thus must be corrected (some earlier than others) for effective information exchange to occur.

Skantze (2007) argues that the error handling functionality in a dialogue system should be the responsibility of all parts in the system, however, certain components may will naturally be adept at handling specific kinds of errors. Errors that cannot be handled in one component may be sent onwards to be corrected by another that is capable (e.g., to the DM which can execute dialogue-level error correction). Lee et al. (2010) makes the claim that error handling approaches in 'traditional DMs' deal with errors through the use of techniques at the conversational level such as with varying degrees of confirmation and rephrasing, implicit and explicit in nature.

In order to accurately plan the conversation and take necessary actions the DM must receive a well-formed input from the NLU component; however it may have information (e.g., context and history) that would otherwise be unavailable to earlier components. Therefore the DM may be given erroneous input from the ASR and NLU components, and still be able to resolve errors using its wider access to dialogue knowledge. Furthermore the DM is able to engage the user in rectifying dialogues, querying them about the authenticity of its beliefs and assumptions about their utterances and even their belief structures.

6. Evaluation of DM Techniques

This report has highlighted a range of important issues that need to be addressed in developing spoken dialogue systems, some key to system functionality while others are peripheral. In this section we discuss the two most important issues we believe are relevant to the development of such systems (§6.1 and §6.2.), and finally evaluate whether such issues have bearing on the Consensus project (§6.3).

6.1 Action Selection Methods

Previously we have established that successful conversation in any dialogue system hinges upon its capability to choose an appropriate response to a user's queries and other multimodal inputs. In our treatment of the issue we segmented the numerous approaches present in the literature into two kinds: handcrafted and machine learning. We briefly described the novelty of each approach, their opportunities and weaknesses, and provided a short summary (§5.3.4) of the two kinds of approaches. Here we continue that discussion and analyse the suitability of approaches with relation to the systems and domains of implementation.

We left §5.3.4 with the conclusion that the domain and circumstances of the system dictate their suitability. We still maintain this observation but clarify the rationale behind our beliefs by recognising that handcrafted and ML systems are distinct, and that the constraints of the intended dialogue system will—in most cases—align most strongly with only one of those.

For instance, the paradigms of handcrafted systems are such that the conversational capabilities they are able to deliver are proportional to the rules and behaviours that have been instilled by human designers and developers. We have shown examples of handcrafted systems that appear to perform dynamically in many situations but they were specifically developed this way. Agent-based systems require the explicit coding of communicative behaviours, FSMs require states and transitions, plan-based approaches require knowledge of the domain and of users and agents (in order to develop plans), and so all are bound by the information they have been given.

This attribute is not necessarily flawed, nor is it undesirable, as there are domains of implementation which are fundamentally predictable. We fall back to the typical scenario of a booking system where the user is only expected to provide answers to questions, slots that the system must fill in before it can give the user a final result (e.g., the booking of transport, a holiday, a task). The literature describing dialogue systems applied in this area is voluminous and this is not unexpected because the inputs and conversations typical of such systems are (mostly) predictable.

We believe the biggest question for deciding between ML and handcrafted approaches is ‘do we *have* the data?’ In supervised learning, the system must be shown a pair: a stimulus (i.e. the user’s input speech) and a desired output. The supervised learning algorithm will attempt to determine the correct function (i.e. mapping) from input to desired output. Data also plays an important role in reinforcement learning where the system is given reinforcements—positive or negative rewards—for its actions.

It is our belief that given a system within a domain wherein training data may be readily available, ML techniques—especially those based on Markovian modelling—may be most appropriate as they have been applied numerously with success in that use case. In specific domains, a data gathering effort could begin where such training data is accumulated for learning. Wizard-of-Oz experiments allow the collection of user interaction data which can be, or add to, training data. Unfortunately, the collection of data may be a large strain on resources which we argue would be infeasible for most.

We believe that exceedingly strict handcraft methods and exceedingly data-reliant ML methods are not suitable for typical domains and realistic constraints of practical systems. However, certain handcrafted methods such as plan-based, agent-based, and frame-based, seem to strike a balance by offering varying degrees of mixed-initiative interaction, whilst not being too reliant on corpora or limited-applicability rules in order to determine what to do next.

6.2 Dialogue with Agents

We spent a couple sections (§5.3.2.6 and §5.4) describing the use of agents for dialogue management and how those agents might be structured within a system. In §4.3.1 we noted that the underlying principle behind the development of ECAs is the closer approximation of human-like conversational abilities, and this lends to the eventual outcome that the user’s interaction with the system is more natural or immersive. For software agents, naturalness and immersion are not as important, as compared to their ability to think and behave rationally with other agents and humans. This is because users expect the system to follow a logically interpret complex situations in the world, and convey that understanding in a way that displays its capacity to interpret and judge those situations.

The ability to display acuity through interpreting and judging situations is one factor influencing user trust, but this expectation should be extended to endow the notion of responsiveness to events in the world as they unfold. This is immediately relevant to the dialogue capability of an agent system, which must handle changes in events as they happen across the duration of conversation.

An agent-based must also be able to recognise its lack of complete understanding of the world and take necessary steps to fill that knowledge gap. Agent approaches to dialogue management actively conscript other agents, software or humans, to generate an answer formed by corporation. Establishing such agreements—ad-hoc or formal in nature—allows a system to form shared knowledge and understanding which could not be realised with an independent agent.

As a final point, we also incorporate the issue of understanding the user with the agent-based architectures due to our observation that both require a modelling component; modelling the user in the former, and modelling *all* agents in the latter. For the reasons discussed in §5.5.2, modelling results in the ability to tailor a user's experience to their personal characteristics and in doing so can gain the trust of that user who can feel confident that the system understands them as an individual.

6.3 Applicability to Consensus

In this section we highlight the most important areas of interest and that we believe are most relevant to the future of Consensus (§6.3.1 and §6.3.2). We base our arguments here upon the findings made in the core of this report and the requirements and scope of Consensus explained earlier in the document (§2). Finally, we make recommendations for forthcoming development based on prioritisations that have been made with regards to the addition of a dialogue management component (§6.3.3).

6.3.1 Choosing an Action Selection Method

We begin with the observation that Consensus, in its current iteration, would benefit from the ability to maintain persistent dialogue with the user, beyond single question-answer pairs—a functionality that is currently absent. This necessarily means that a dialogue manager (DM) must be present, as it was established that maintaining conversation is one of the key roles it fills within a spoken dialogue system. Initial discussion regarding a DM should keep action selection at its core, as this is the responsibility that truly defines how the interaction is calculated. The DM must choose between two methods of action selection, ML or handcrafted, and each have their pros and cons as we have covered extensively (§5.1 and §6.1). We take the general issues we have highlighted with those methods and apply them directly to the needs of Consensus.

Dialogue extensibility is the ability of the DM to adopt new methods of action selection, engage in dialogues that were not initially included with the component. For ML systems, this is achieved primarily through two methods: reinforcement or supervised learning upon training data, and the same learning algorithms applied online to actual interaction with users. Making such alterations with handcrafted systems is not automatic and requires that the system's experts apply their time to identify extensions to current dialogue strategies. As the number of strategies increase, so does the complexity of the DM as a whole, resulting in more time needed to ensure additions conflate, but do not conflict with, previous strategies. The Consensus project has extensibility as one desirable factor, and the effort and resources that would be required to extend handcrafted approaches may be undesirable in a system expected or intended to grow over time.

We distinguish between two characteristics of DMs relevant to their domain knowledge capacities. Intra-domain extensibility is the ability to add new understanding of objects, concepts, and relationships within the same domain. Inter-domain portability is similar but refers to learning about a completely different domain. ML systems deal well with both, but applying them to new domains requires additional datasets. Handcrafted systems are much more difficult, and the line between what is the same domain or a different domain becomes blurred. In the case of either approach to action selection, subject matter experts are required for any domain additions. It is intended that Consensus be able to expand its domain knowledge over time, so we must consider how this could be achieved with respect to the choice made of dialogue management approach.

We have established that ML approaches to dialogue management require a dataset consisting of user inputs, which are paired with an expected output, reward, or cost. For Consensus within its specific military domain scenario, the question of how such pairings are acquired is a pertinent one given a large amount would be required. Wizard-of-Oz experiments have been cited in this report as a means by which ML-based systems gather datasets by having a human experimenter act as though the system should; thus input (user utterances) and output (system utterances) are accumulated with this interaction. This description obfuscates that the human experimenter must have some knowledge of how the system *should* act, essentially making them a subject matter expert of sorts, often meaning that a predetermined *script* must have been developed prior.

Even if the data can be collated through some means—such as Wizard-of-Oz experiments—then there still must be some form of evaluation to evaluate the correctness and validity of the action selection methods after they have been learnt via the ML algorithms. In fact the same can be said of handcrafted approaches which, although tend to be more deterministic, their rules, plans, FSMs, or schemas must also be verified with test user interactions.

An important question to ask is whether there exist clear methods for the collection of domain and interaction data for ML, and whether it is feasible for experts to encode the domain using the paradigms of a handcrafted approach. Then, finally, which of those two alternatives are practically available to the future development efforts of Consensus?

Because handcrafted and ML systems are fundamentally different they require different kinds of experts: the former rely on linguists and speech theorists to develop the necessary dialogue strategies used by the DM, whilst the latter requires experts specialising in the use of ML systems to develop the appropriate algorithms for action selection. It should be noted that in either scenario software developers are clearly required for software implementation, and subject matter experts are needed to verify that the system behaves according to its prescribed requirements and domain scenario.

Consensus conducts complex situational awareness and as part of its reasoning capacity it requires a high degree of verifiability—a transparency where an outside observer can trace the inputs given to the system, and the system's outputs. A DM processes input utterances and produces a system act; the way that a handcrafted system conducts this processing is different to ML systems. In the former: rules, plans, or states can be traced to find how the DM led to a particular conclusion (the system response) given a particular input—we summarise them as a *white box*. In the latter, ML algorithms are less transparent to outside observers and essentially appear as *black boxes* where the processing cannot be understood.

We do not suggest that just because some parts of Consensus have transparency and verifiability then so should the components of dialogue management. Instead we clarify that the DM should be modularised and free to operate within its own paradigms, independent of the rest of the system.

6.3.2 Identifying the User

As a complex situational awareness system, Consensus must be able to deliver information appropriate to a user's needs at a given time. Through identifying those needs, the system will be able to provide relevancy to the user's interaction—for reasons explained in §5.5.1. It may be noted that robust methods of action selection do not necessarily imply the ability of a dialogue manager to accurately attend to the user's intentions—they simply map an event (e.g., the user's utterance) to an action to take. Of course it may be that Consensus is concerned with explicitly stated utterances where the intent is clear and unambiguous. However, if it is to grow and begin to accept some of the more abstruse statements of intent then there must be an additional functionality here for that purpose.

We identified user modelling (§5.5.2) as an important issue for dialogue management, as it permits a better understanding of who the user is—thus better understanding what they want; this links very well to the notion of intent discussed above. User models, depending upon implementation specifics, can be immensely valuable in all areas of dialogue management and open opportunities for customised behaviour based on a single user or an entire cohort of users. As a complex situational awareness system it is important for the system to convey its own understanding of the world to the user in a form the user can comprehend. The different users of Consensus—with different roles, ranks, and personalities—will naturally interact with it in different ways and have preferences about what information they are given and how it is displayed.

We believe that early adoption of user modelling techniques—taken advantage of for the purposes of dialogue management—within Consensus will be beneficial to understanding the user beyond their utterances, and create an experience that can be tailored to explicitly appeal to who the user is and what they want.

6.3.3 Recommendations

A DM is a complex component and carries out many functions, and it would be imprudent to have the intent to develop such an artefact in its entirety from the outset. In planning for a future DM component within Consensus, a prioritisation was made of desired dialogue capabilities and a roadmap was conceived with the hope of identifying the low hanging fruit—the aspects of dialogue that are most important to the current iteration of Consensus. Here we provide a list of some of those key aspects as relevant to the recommendations we make below:

- Use of dialogue strategies and other related capabilities including turn-taking, error handling strategies, and content generation methods (e.g., summarisation and confirmation).
- Understanding of the intents, goals, and other characteristics of the user (e.g., their role, preferences, and domain knowledge) in order to tailor the dialogue interaction.
- Modular and interoperable development to achieve flexibility and extensibility.

Throughout this report we have emphasised the role of action selection as one of the primary concerns of a DM, and indicated the variety of approaches available to achieve this. Before Consensus is to adopt one of these approaches, an effort must be made to model the dialogue strategies that are desired of the system. Strategies dictating the system's methodology for handling errors (and other such miscommunications) have been evaluated as being critical in the early stages of development; Consensus already possesses a natural language interface but still requires an ability to cope with unexpected inputs.

Discussions have taken place that identify a need to represent the Consensus system as an agent, one of many that would form a multi-agent architecture as we have described in this report. A multi-agent architecture is a positive step and highly desirable given that Consensus must handle multiple domains within a military scenario, but this architecture obscures the still-relevant need to decide upon a method of action selection. Agents interact with each other via the use of collaborative protocols, but internally must also maintain a method of action selection that determines how that particular agent will choose what it says.

Specific and detailed use cases can be written that help guide the decision for the adoption of a particular approach by identifying patterns of dialogue which may be achievable more in one approach than in another. For instance, a sequential and decision-based dialogue example may indicate suitability for a finite-state dialogue model. A clear recommendation for a single approach cannot be made in this report, and we defer such a decision until the requirements of dialogue are made explicit. Instead we conclude that there are many approaches with their associated pros and cons—including a varying degree of expressivity and flexibility—and believe use cases will help identify which will be useful.

Use cases can also help identify tasks that align with components already present in the existing infrastructure, promoting the use of existing architecture for some actions. Indeed, a primary concern for a DM component in Consensus is that it is interoperable with the infrastructure currently present. We believe it would be advantageous to identify *interfaces* (e.g., application programming interfaces) that define what operations Consensus and a prospective DM should offer to each other; and indeed to other components. Interfaces are common practice for program design and development and should thus be front and centre in the discussion for interoperability. ML and handcrafted approaches to action selection are certainly distinct, but in the end they just return an action based on the user's input; interfaces should capitalise upon this and other such observations.

The discussion made thus far has been with the assumption that a dialogue manager would be created from scratch in order to complete the spoken dialogue interface within Consensus; this is not the only option. Throughout this report we have presented several academic systems and toolkits which may be available to Defence and which may be adopted for experimentation; the benefits to this approach are obvious and will not be discussed. It may be worth testing implementations such as *TrindiKit* which have been employed in many other systems and have a bounty of literature available to study. Whilst we noted before that ML techniques should not be overlooked, there certainly are not very many well-known toolkits for their development and their availability must also be considered in addition to the intrinsic benefits or costs associated with the techniques themselves.

Consideration must also be awarded to the fact that it would be beneficial for any SDS to understand its user through user modelling and intent recognition. The recognition of intents and modelling of the user are two features that have much in common and can plausibly be developed in unison. Before development of either begins, however, an in-depth understanding of the intended users must be sought to guide development through the understanding of expected typical usage, interaction habits, task-related considerations, and actual needs of the users. Earlier we discussed the importance of use cases for the development of dialogue strategies to inform action selection decisions, and we contend here also that they may garner a greater understanding of the kinds of interactions to focus upon.

In this section we have given some basic recommendations and considerations that will define future design and development for dialogue management within Consensus. Attention has been paid to some preliminary actions that should occur to precisely scope the solution, including the dialogue expectations and intended users. A multi-agent architecture has been proposed as a suitable paradigm for Consensus as a complex situational awareness system, promoting its appearance as a trustworthy system and facilitating flexibility in dialogue management. We have made the case for defining interfaces to facilitate the desired interoperability of current and future components, and explained possible benefits for venturing into extant academic toolkits and systems. Finally, the development of detailed use cases is currently underway but is outside the scope of this report.

7. Conclusion

To create a spoken dialogue system (SDS) with a conversational prowess equivalent to that of a human remains a seemingly out-of-reach problem. We, as humans, are able to integrate different kinds of contextual knowledge, and years of experience, to guide our decisions and hypotheses about the dialogue. Machines can attempt similar inferencing with statistics, machine learning, and various models of discourse and context, but no present system has displayed the degree of competence of human dialogue. In this report we have discussed some key areas of interest for spoken dialogue systems and dialogue management. We also provided insight and recommendations for their application to the Consensus project.

A substantial body of work over the decades has been produced in refining SDS technology, but has yet to identify a single universal solution for design and implementation. Many theoretical proposals exist, some of which have been covered in this report, but it is clear that there is no single solution to how spoken dialogue systems should be developed. Future development of Consensus should carefully prioritise the aspects of dialogue and discourse that are most important—a process which has already begun—with due consideration of the latest research in the field.

The push for embodied conversational agents has also led to research into creating realistic and engaging systems with the aim to resemble human-like interaction. We believe further advancements will make the interface to complex automated systems utilise human communicative strategies. These techniques have broad application in automated systems such as in the following use cases:

- Control: disability support, personal assistance
- Education: tutorial and training
- Medical: patient monitoring, assessment and diagnosis
- Military: command and control, situation awareness
- Monitoring and Alert: ongoing status updates
- Social: companionship, advice, entertainment.

Further, it is inevitable that SDSs will become more sophisticated and powerful over the years; the advancements already observed are due to innovative efforts to combine dialogue systems with new methodologies. By taking advantage of novel or upcoming techniques and applying them to dialogue systems, new paradigms may be achieved; we have already observed this with the use of machine learning DMs and even multi-agent architectures. We may be a long way off before HAL is developed, but we look forward to seeing improvement in spoken dialogue systems in the future.

8. References

Allen, J. F., & Perrault, C. R. (1980). Analyzing intention in utterances. *Artificial intelligence, 15*(3), pp. 143-178.

Allen, J. F., & Schubert, L. K. (1991). The TRAINS Project *TRAINS Technical Note*. Rochester, NY: Department of Computer Science, University of Rochester.

Bhargava, A., Celikyilmaz, A., Hakkani-Tür, D., & Sarikaya, R. (2013). *Easy contextual intent prediction and slot detection.* In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, Canada, pp. 8337-8341. IEEE.

Bohus, D., Raux, A., Harris, T. K., Eskenazi, M., & Rudnicky, A. I. (2007). *Olympus: an open-source framework for conversational spoken language interface research.* In Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies, Rochester, New York, pp. 32-39. Association for Computational Linguistics.

Bohus, D., & Rudnicky, A. I. (2003). *RavenClaw: Dialog management using hierarchical task decomposition and an expectation agenda.* In Proceedings of the Eurospeech Conference on Speech, Communication and Technology, Geneva, Switzerland, pp. 597-600.

Bohus, D., & Rudnicky, A. I. (2009). The RavenClaw dialog management framework: Architecture and systems. *Computer Speech & Language, 23*(3), pp. 332-361.

Boye, J. (2007). *Dialogue management for automatic troubleshooting and other problem-solving applications.* In Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, Antwerp, Belgium, pp. 247-255.

Bratman, M. E., Israel, D. J., & Pollack, M. E. (1988). Plans and resource-bounded practical reasoning. *Computational intelligence, 4*(3), pp. 349-355.

Bretier, P., & Sadek, D. (1996) A rational agent as the kernel of a cooperative spoken dialogue system: Implementing a logical theory of interaction. *Vol. 1193. Lecture Notes in Computer Science* (pp. 189-203): Springer Verlag.

Bunt, H. (1994). Context and dialogue control. *Think Quarterly, 3,* pp. 19-31.

Callan, R. (2003). *Artificial Intelligence*. New York, NY: Palgrave MacMillan, ISBN 0-333-80136-9.

Callejas, Z., Griol, D., & López-Cózar, R. (2011). Predicting user mental states in spoken dialogue systems. *EURASIP Journal on Advances in Signal Processing, 2011*(1), pp. 1.

Cassell, J. (2000). Embodied conversational interface agents. *Communications of the ACM, 43*(4), pp. 70-78.

Cassell, J. (2001). Embodied conversational agents: representation and intelligence in user interfaces. *AI magazine, 22*(4), pp. 67-83.

Cassell, J., Bickmore, T., Campbell, L., & Vilhjálmsson, H. (2000). "Human conversation as a system framework: Designing embodied conversational agents", *Embodied conversational agents* (pp. 29-63). Boston: MIT Press, ISBN 9780262032780.

Chien, J.-T., & Chueh, C.-H. (2012). Topic-based hierarchical segmentation. *IEEE Transactions on Audio, Speech, and Language Processing, 20*(1), pp. 55-66.

Chu-Carroll, J., & Carberry, S. (1994). *A plan-based model for response generation in collaborative task-oriented dialogues.* In Proceedings of the 12th National Conference on Artificial Intelligence, Seattle, Washington, pp. 799-905.

Commonwealth of Australia, Department of the Prime Minister and Cabinet (Australia). *Strong and secure: a strategy for Australia's national security*. Retrieved 19 September 2016, from http://apo.org.au/node/33996.

Commonwealth of Australia, First Principles Review Team. *First Principles Review - Creating One Defence*. Retrieved 26 October 2016, from http://www.defence.gov.au/Publications/Reviews/Firstprinciples/Docs/FirstPrinciplesReviewB.pdf.

Commonwealth of Australia, Department of Defence. *2016 Defence White Paper*. Retrieved 19 September 2016, from http://apo.org.au/node/61805.

DeVault, D., Mell, J., & Gratch, J. (2015). *Toward natural turn-taking in a virtual human negotiation agent.* In Proceedings of the AAAI Spring Symposium on Turn-taking and Coordination in Human-Machine Interaction, Stanford, California. AAAI Press.

Dragone, P. (2015). *Non-Sentential Utterances in Dialogue: Experiments in classification and interpretation.* In Proceedings of the 19th Workshop on the Semantics and Pragmatics of Dialogue, Gothenburg, Sweden, pp. 170-171. Gothenburg University.

Edlund, J., Skantze, G., & Carlson, R. (2004). *Higgins - a spoken dialogue system for investigating error handling techniques.* In Proceedings of the 8th International Conference on Spoken Language Processing, Jeju, Korea.

Endsley, M. R. (1988). *Design and evaluation for situation awareness enhancement.* In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Anaheim, California, pp. 97-101. SAGE Publications.

Ferguson, G., & Allen, J. F. (2011). *A Cognitive Model for Collaborative Agents.* In Proceedings of the AAAI 2011 Fall Symposium on Advances in Cognitive Systems, Washington, D.C, pp. 112-120. Elsevier.

Fischer, G. (2001). User modeling in human–computer interaction. *User modeling and user-adapted interaction, 11*(1-2), pp. 65-86.

Flycht-Eriksson, A. (2001). *Domain knowledge management in information-providing dialogue systems.* (Licentiate of Philosophy Doctoral dissertation), Linköping University. Linköping University.

Graesser, A. C., Franklin, S., Wiemer-Hastings, P. M., & Group, T. R. (1998, May 1998). *Simulating Smooth Tutorial Dialogue with Pedagogical Value.* In Proceedings of the FLAIRS Conference, Sanibel Island, California, pp. 163-167.

Griol, D., Iglesias, J. A., Ledezma, A., & Sanchis, A. (2016). A Two-Stage Combining Classifier Model for the Development of Adaptive Dialog Systems. *International journal of neural systems, 26*(01), pp. 1650002.

Grosz, B. J., & Sidner, C. L. (1986). Attention, intentions, and the structure of discourse. *Computational linguistics, 12*(3), pp. 175-204.

Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of computer programming, 8*(3), pp. 231-274.

Higashinaka, R., Sudoh, K., & Nakano, M. (2006). Incorporating discourse features into confidence scoring of intention recognition results in spoken dialogue systems. *Speech Communication, 48*(3), pp. 417-436.

Hofstadter, D., & Boden, M. A. (1995). "Preface 4 The Ineradicable Eliza Effect and Its Dangers,Epilogue", *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. New York: Basic Books, ISBN 978-0-465-02475-9.

Hong, J.-H., & Cho, S.-B. (2003). *A two-stage Bayesian network for effective development of conversational agent.* In Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning, Hong Kong, pp. 1-9. Springer.

Horacek, H., & Wolska, M. (2005). *A hybrid model for tutorial dialogs.* In Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue, Lisbon, Portugal, pp. 190-199. Elsevier.

Horvitz, E., Breese, J., Heckerman, D., Hovel, D., & Rommelse, K. (1998). *The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users.* In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, Madison, Wisconsin, pp. 256-265. Morgan Kaufmann Publishers Inc.

Hu, B., Lu, Z., Li, H., & Chen, Q. (2014). *Convolutional neural network architectures for matching natural language sentences.* In Proceedings of the Advances in Neural Information Processing Systems, Montréal, Canada, pp. 2042-2050.

Jennings, N. R., Sycara, K., & Wooldridge, M. (1998). A roadmap of agent research and development. *Autonomous agents and multi-agent systems, 1*(1), pp. 7-38.

Johansson, P. (2002). User modeling in dialog systems. *St. Anna Report SAR*, pp. 02-02.

Jonson, R. (2006). *Dialogue context-based re-ranking of ASR hypotheses.* In Proceedings of the 2006 IEEE Spoken Language Technology Workshop, Palm Beach, Aruba, pp. 174-177. IEEE.

Jurafsky, D., & Martin, J. H. (2009). *Speech and Language Processing* (Pearson International 2nd ed.). Upper Saddle River, N.J.: Pearson/Prentice Hall, ISBN 978-0-13-504196-3.

Jurčíček, F., Thomson, B., & Young, S. (2011). Natural actor and belief critic: Reinforcement algorithm for learning parameters of dialogue systems modelled as POMDPs. *ACM Transactions on Speech and Language Processing (TSLP), 7*(3), pp. 6.

Kass, R., & Finin, T. (1988). *Modeling the User in Natural Language Systems*: Springer Science & Business Media, ISBN 364283230X.

Kobsa, A. (1993). User modeling: Recent work, prospects and hazards. *Human Factors in Information Technology, 10*, pp. 111-111.

Kobsa, A. (2001). Generic user modeling systems. *User modeling and user-adapted interaction, 11*(1-2), pp. 49-63.

Kronlid, F. (2006). *Turn taking for artificial conversational agents.* In Proceedings of the International Workshop on Cooperative Information Agents, Edinburgh, UK, pp. 81-95. Springer.

Lambert, D. A., Saulwick, A., & Trentelman, K. (2015, 2015). *Consensus: A comprehensive solution to the grand challenges of information fusion.* In Proceedings of the 18th International Conference on Information Fusion, Washington, D.C, pp. 908-915. Institute of Electrical and Electronics Engineers Inc.

Larsson, S. (2002). *Issue-based dialogue management*: Department of Linguistics, Göteborg University, ISBN 9162857355.

Larsson, S., & Traum, D. R. (2000). Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural language engineering, 6*(3&4), pp. 323-340.

Lee, C., Jung, S., Eun, J., Jeong, M., & Lee, G. G. (2006). *A situation-based dialogue management using dialogue examples.* In Proceedings of the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing, Toulouse, France, pp. 69-72. IEEE.

Lee, C., Jung, S., Kim, K., Lee, D., & Lee, G. G. (2010). Recent approaches to dialog management for spoken dialog systems. *Journal of Computing Science and Engineering, 4*(1), pp. 1-22.

Lee, J., & Marsella, S. (2006). *Nonverbal behavior generator for embodied conversational agents.* In Proceedings of the International Workshop on Intelligent Virtual Agents, Marina Del Rey, California, pp. 243-255. Springer.

Lee, S.-I., Sung, C., & Cho, S.-B. (2001). *An effective conversational agent with user modeling based on Bayesian network.* In Proceedings of the Web Intelligence: Research and Development, Maebashi City, Japan, pp. 428-432. Springer.

Lemon, O. (2011). Learning what to say and how to say it: Joint optimisation of spoken dialogue management and natural language generation. *Computer Speech & Language, 25*(2), pp. 210-221.

Leuski, A., & Traum, D. (2008, December 2008). *A statistical approach for text processing in virtual humans.* In Proceedings of the 26th Army Science Conference, Orlando, FL, pp. 8.

Levin, E., Pieraccini, R., & Eckert, W. (2000). A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on speech and audio processing, 8*(1), pp. 11-23.

Lim, S., Oh, K., & Cho, S.-B. (2010). *A Spontaneous Topic Change of Dialogue for Conversational Agent Based on Human Cognition and Memory.* In Proceedings of the International Conference on Agents and Artificial Intelligence, Valencia, Spain, pp. 91-100. Springer.

Lin, B.-s., Wang, H.-m., & Lee, L.-s. (1999). *A distributed architecture for cooperative spoken dialogue agents with coherent dialogue state and history.* In Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding

Lison, P. (2015). A hybrid approach to dialogue management based on probabilistic rules. *Computer Speech & Language, 34*(1), pp. 232-255.

LuperFoy, S., Loehr, D., Duff, D., Miller, K., Reeder, F., & Harper, L. (1998). *An architecture for dialogue management, context tracking, and pragmatic adaptation in spoken dialogue systems.* In Proceedings of the 36th ACL and the 17th ACL-COLING, Montreal, Canada, pp. 794-801. Association for Computational Linguistics.

McTear, M. F. (2002). Spoken dialogue technology: enabling the conversational user interface. *ACM Computing Surveys (CSUR), 34*(1), pp. 90-169.

Meng, F., Lu, Z., Tu, Z., Li, H., & Liu, Q. (2015). *A Deep Memory-based Architecture for Sequence-to-Sequence Learning.* In Proceedings of the ICLR Workshop, San Juan, Puerto Rico.

Mingxuan, W., Zhengdong, L., Li, H., Jiang, W., & Liu, W. J. Q. (2015). *A Convolutional Architecture for Word Sequence Prediction.* In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, pp. 9.

Moore, J. D., & Paris, C. L. (1993). Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational linguistics, 19*(4), pp. 651-694.

Moore, R. C., Dowding, J., Bratt, H., Gawron, J. M., Gorfu, Y., & Cheyer, A. (1997). *CommandTalk: A spoken-language interface for battlefield simulations.* In Proceedings of the 5th Conference on Applied Natural Language Processing, Washington, D.C, pp. 1-7. Association for Computational Linguistics.

Morbini, F., DeVault, D., Sagae, K., Gerten, J., Nazarian, A., & Traum, D. (2014). "FLoReS: a forward looking, reward seeking, dialogue manager", *Natural Interaction with Robots, Knowbots and Smartphones* (pp. 313-325). Springer Link: Springer, ISBN 978-1-4614-8280-2.

Mouromtsev, D., Kovriguina, L., Emelyanov, Y., Pavlov, D., & Shipilo, A. (2015). *From spoken language to ontology-driven dialogue management.* In Proceedings of the International Conference on Text, Speech, and Dialogue, Pilsen, Czech Republic, pp. 542-550. Springer International Publishing.

Moussa, M. B., Kasap, Z., Magnenat-Thalmann, N., Chandramouli, K., Haji Mirza, S. N., Zhang, Q., Izquierdo, E., Biperis, I., & Daras, P. (2010). *Towards an expressive virtual tutor: an implementation of a virtual tutor based on an empirical study of non-verbal behaviour.* In Proceedings of the 2010 ACM Workshop on Surreal Media and Virtual Cloning, Firenze, Italy, pp. 39-44. ACM.

Nass, C., Isbister, K., & Lee, E.-J. (2000). Truth is beauty: Researching embodied conversational agents. *Embodied conversational agents*, pp. 374-402.

Novielli, N., de Rosis, F., & Mazzotta, I. (2010). User attitude towards an embodied conversational agent: Effects of the interaction mode. *Journal of Pragmatics, 42*(9), pp. 2385-2397.

Nye, B. D., Graesser, A. C., & Hu, X. (2014). AutoTutor and family: A review of 17 years of natural language tutoring. *International Journal of Artificial Intelligence in Education, 24*(4), pp. 427-469.

Papangelis, A., Karkaletsis, V., & Makedon, F. (2012a). *Online complex action learning and user state estimation for adaptive dialogue systems.* In Proceedings of the 24th IEEE International Conference on Tools with Artificial Intelligence, Piraeus, Greece, pp. 642-649. IEEE.

Papangelis, A., Kouroupas, N., Karkaletsis, V., & Makedon, F. (2012b). *An Adaptive Dialogue System with Online Dialogue Policy Learning.* In Proceedings of the 7th Hellenic Conference on Artifical Intelligence, Lamia, Greece, pp. 323-330. Springer.

Papangelis, A., Makedon, F., & Gatchel, R. (2014). Assessing and Monitoring Post-Traumatic Stress Disorder Through Natural Interaction With an Adaptive Dialogue System. *Journal of Applied Biobehavioral Research, 19*(3), pp. 192-215.

Pelachaud, C. (2005). *Multimodal expressive embodied conversational agents.* In Proceedings of the 13th Annual ACM International Conference on Multimedia, Singapore, pp. 683-689. ACM.

Perrault, C. R., Allen, J. F., & Cohen, P. R. (1978). *Speech acts as a basis for understanding dialogue coherence.* In Proceedings of the Workshop on Theoretical Issues in Natural Language Processing, Urbana, USA, pp. 125-132. Association for Computational Linguistics.

Raux, A. (2008). *Flexible turn-taking for spoken dialog systems.* US National Science Foundation.

Raux, A., & Eskenazi, M. (2012). Optimizing the turn-taking behavior of task-oriented spoken dialog systems. *ACM Transactions on Speech and Language Processing (TSLP), 9*(1), pp. 1.

Rich, E. (1979). User modeling via stereotypes. *Cognitive science, 3*(4), pp. 329-354.

Rickel, J., & Johnson, W. L. (2000). Task-oriented collaboration with embodied agents in virtual worlds. *Embodied conversational agents*, pp. 95-122.

Riviere, J., Adam, C., Pesty, S., Pelachaud, C., Guiraud, N., Longin, D., & Lorini, E. (2011). *Expressive multimodal conversational acts for SAIBA agents.* In Proceedings of the 13th International Conference on Intelligent Virtual Agents, Reykjavik, Iceland, pp. 316-323. Berlin, Germany: Springer Verlag.

Rudnicky, A., Thayer, E. H., Constantinides, P. C., Tchou, C., Shern, R., Lenzo, K. A., Xu, W., & Oh, A. (1999). *Creating natural dialogs in the carnegie mellon communicator system.* In Proceedings of the 6th European Conference on Speech Communication and Technology, Budapest, Hungary, pp. 1531-1534.

Rudnicky, A., & Xu, W. (1999). *An agenda-based dialog management architecture for spoken language systems.* In Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop, Seattle, WA.

Russell, S. J., & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach* (3rd ed.). Upper Saddle River, NJ: Prentice Hall, ISBN 978-0-13-604259-4.

Sacks, H., Schegloff, E. A., & Jefferson, G. (1974). A simplest systematics for the organization of turn-taking for conversation. *Language 50*(4), pp. 696-735.

Schröder, M. (2010). The SEMAINE API: towards a standards-based framework for building emotion-oriented systems. *Advances in human-computer interaction, 2010*, pp. 319-406. DOI: 10.1155/2010/319406

Schubert, L. K. (2005). *Some knowledge representation and reasoning requirements for self-awareness.* In Proceedings of the AAAI Spring Symposium on Metacognition in Computation, Palo Alto, CA, USA, pp. 106-113.

Searle, J. R. (1969). *Speech acts: An essay in the philosophy of language* (Vol. 626): Cambridge university press, ISBN 052109626X.

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., & Pineau, J. (2016). *Building end-to-end dialogue systems using generative hierarchical neural network models.* In Proceedings of the 30th AAAI Conference on Artificial Intelligence, Phoenix, Arizona, pp. 3776-3783.

Shang, L., Lu, Z., & Li, H. (2015). *Neural responding machine for short-text conversation.* In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, Beijing, China, pp. 1577-1586.

Skantze, G. (2005). *Galatea: A discourse modeller supporting concept-level error handling in spoken dialogue systems.* In Proceedings of the 6th SIGDIAL Workshop on Discourse and Dialogue, Lisbon, Portugal, pp. 178-189. Springer.

Skantze, G. (2007). *Error Handling in Spoken Dialogue Systems-Managing Uncertainty, Grounding and Miscommunication.* (Doctoral Thesis in Speech Communication), KTH Royal Institute of Technology. Stockholm, Sweden.

Skantze, G., & Edlund, J. (2004). *Robust interpretation in the Higgins spoken dialogue system.* In Proceedings of the COST278 and ISCA Tutorial and Research Workshop on Robustness Issues in Conversational Interaction, Nowich, UK. ISCA.

Sleeman, D. (1985). UMFE: a user modelling front-end subsystem. *International Journal of Man-Machine Studies, 23*(1), pp. 71-88.

Smith, C., Crook, N., Dobnik, S., Charlton, D., Boye, J., Pulman, S., De La Camara, R. S., Turunen, M., Benyon, D., & Bradley, J. (2011). Interaction strategies for an affective conversational agent. *Presence: Teleoperators and Virtual Environments, 20*(5), pp. 395-411.

Sonntag, D. (2006). *Towards combining finite-state, ontologies, and data driven approaches to dialogue management for multimodal question answering.* In Proceedings of the 5th Slovenian First International Language Technology Conference, Ljubljana, Slovenia, pp. 210-215.

Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.-Y., Gao, J., & Dolan, B. (2015). *A neural network approach to context-sensitive generation of conversational responses.* In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, pp. 196-205.

Stent, A., Dowding, J., Gawron, J. M., Bratt, E. O., & Moore, R. (1999). *The CommandTalk Spoken Dialogue System.* In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, College Park, Maryland, pp. 183-190. Association for Computational Linguistics.

Traum, D., Bos, J., Cooper, R., Larsson, S., Lewin, I., Matheson, C., & Poesio, M. (1999). A model of dialogue moves and information state revision (Department of Linguistics) (pp. 86). Gothenburg: Gothenburg University.

Van Noord, G., Bouma, G., Koeling, R., & Nederhof, M.-J. (1999). Robust grammatical analysis for spoken dialogue systems. *Natural language engineering, 5*(01), pp. 45-93.

Wahlster, W., & Kobsa, A. (1989). *User models in dialog systems*: Springer, ISBN 3642832326.

Walker, M. A., Litman, D. J., Kamm, C. A., & Abella, A. (1997). *PARADISE: A framework for evaluating spoken dialogue agents.* In Proceedings of the 8th Conference on European Chapter of the Association for Computational Linguistics, Madrid, Spain, pp. 271-280. Association for Computational Linguistics.

Wang, K. (2000). *A plan-based dialog system with probabilistic inferences.* In Proceedings of the INTERSPEECH Annual Conference, Beijing, China, pp. 644-647.

Webb, N. *Rule-based dialogue management systems.* (2000) [Accessed 26 October 2016]; Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.22.2854&rep=rep1&type=pdf, DOI: 10.1.1.22.2854.

Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM, 9*(1), pp. 36-45.

White, F. E. (1988). *A model for data fusion.* In Proceedings of the 1st National Symposium on Sensor Fusion, pp. 149-158.

Wiemer-Hastings, P., Graesser, A. C., Harter, D., & Group, T. R. (1998). *The foundations and architecture of AutoTutor.* In Proceedings of the International Conference on Intelligent Tutoring Systems, San Antonio, Texas, pp. 334-343. Springer.

Williams, J. D., & Young, S. (2007). Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language, 21*(2), pp. 393-422.

Wollmer, M., Schuller, B., Eyben, F., & Rigoll, G. (2010). Combining long short-term memory and dynamic bayesian networks for incremental emotion-sensitive artificial listening. *IEEE Journal of Selected Topics in Signal Processing, 4*(5), pp. 867-881.

Wooldridge, M. (2009). *An introduction to multiagent systems*: John Wiley & Sons, ISBN 0470519460.

Wu, X., Zheng, F., & Xu, M. (2001). *Topic forest: A plan-based dialog management structure.* In Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, Salt Lake City, Utah, pp. 617-620. IEEE.

Xu, Q., Li, L., & Wang, G. (2013). *Designing engagement-aware agents for multiparty conversations.* In Proceedings of the 31st Annual CHI Conference on Human Factors in Computing Systems, Paris, France, pp. 2233-2242. ACM.

Yankelovich, N., & Baatz, E. (1994). *SpeechActs: A Framework for Building Speech Applications.* In Proceedings of the American Voice I/O Society Conference, San Jose, California, pp. 20-23. Citeseer.

Young, S., Schatzmann, J., Weilhammer, K., & Ye, H. (2007). *The hidden information state approach to dialog management.* In Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing, Honolulu, Hawaii, pp. 149-152. IEEE.

Zukerman, I., & Litman, D. (2001). Natural language processing and user modeling: Synergies and limitations. *User modeling and user-adapted interaction, 11*(1-2), pp. 129-158.

Appendix A: Mentioned Systems

System	Description	Page	Reference
ARTIMIS	A rational agent based upon the implementation of a formal theory of interaction.	27	(Bretier & Sadek, 1996)
AutoTutor	A pedagogically focused system which asks the human learner a series of questions about a domain using production rules.	10-11	(Graesser et al., 1998; Wiemer-Hastings et al., 1998)
CMU Communicator	A frame-based dialogue system in an itinerary generating domain that served as the precursor to *Olympus*.	11	(Rudnicky et al., 1999)
CommandTalk	A spoken interface to a battlefield simulation system using an agent architecture and finite-state dialogue management; developed by SRI International.	11	(Moore et al., 1997; Stent et al., 1999)
Consensus	Complex situational awareness system developed by Australia's DST Group.	2-3, 39-43	(Lambert et al., 2015)
Cortana	Intelligent personal assistant by Microsoft.	11	See footnote[17]
ELIZA	A chat-bot by Weizenbaum whose primary purpose was to behave and respond as a Rogerian psychologist through pattern matching and transformations.	9-10, 16, 44	(Weizenbaum, 1966)
GALATEA	An information state-based discourse modeller and component of the *HIGGINS* spoken dialogue system.	18	(Edlund et al., 2004; Skantze, 2005, 2007; Skantze & Edlund, 2004)
Google Now	Intelligent personal assistant by Google.	11	See footnote[18]

[17] https://www.microsoft.com/en-us/mobile/experiences/cortana/ . Accessed 31 October 2016.
[18] https://www.google.com/search/about/learn-more/now/ . Accessed 31 October 2016.

GRUNDY	Intelligent (non-spoken dialogue) system that utilises stereotypes and understanding of users to deliver customised book recommendations; minimal textual dialogue capability.	33	(Rich, 1979)
HAL 9000	Fictional rogue AI and primary antagonist in the screenplay for 2001: A Space Odyssey, portraying a sophisticated dialogue as well as intelligence capability.	10	See footnote[19]
Olympus	Spoken dialogue system developed by Carnegie Mellon University, a successor to the *CMU Communicator* project.	15, 25	(Bohus et al., 2007)
OpenDial	A rule-based system which utilises probabilistic aspects to select rules based on their likelihood.	19	(Lison, 2015)
RavenClaw	The frame- or agenda-based dialogue manager within the *Olympus* spoken dialogue system that structures discourse as a hierarchy of frames, enabling mixed-initiative dialogue.	11, 20, 25	(Bohus & Rudnicky, 2003, 2009)
SEMAINE	An SDS offering chit-chat behaviour with a clear focus on emotion recognition as well as production through an embodied agent with facial expressions and tone of voice.	11-12, 14	(Schröder, 2010)
Siri	Intelligent personal assistant by Apple.	11	See footnote[20]
SpeechActs	A research prototype framework "for building and integrating multiple speech applications".	14	(Yankelovich & Baatz, 1994)
UMFE	A User Modelling Front-End subsystem that acts as an interface between a user and an intelligent system.	33	(Sleeman, 1985)

[19] http://www.imdb.com/title/tt0062622/ . Accessed 31 October 2016.
[20] http://www.apple.com/au/ios/siri/ . Accessed 31 October 2016.

DEFENCE SCIENCE AND TECHNOLOGY GROUP DOCUMENT CONTROL DATA

1. DLM/CAVEAT (OF DOCUMENT)

2. TITLE

Dialogue Systems & Dialogue Management

3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (U/L) NEXT TO DOCUMENT CLASSIFICATION)

Document	(U)
Title	(U)
Abstract	(U)

4. AUTHOR(S)

Deeno Burgan

5. CORPORATE AUTHOR

DST Group Edinburgh
PO Box 1500
Edinburgh SA 5111

6a. DST Group NUMBER DST-Group-TR-3331	6b. AR NUMBER AR-016-776	6c. TYPE OF REPORT Technical Report	7. DOCUMENT DATE December 2016

8. Objective ID AV14615545	9. TASK NUMBER	10. TASK SPONSOR

13. DOWNGRADING/DELIMITING INSTRUCTIONS

14. RELEASE AUTHORITY
Chief, National Security & Intelligence, Surveillance, and Reconnaissance Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release.

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111

16. DELIBERATE ANNOUNCEMENT

No limitations.

17. CITATION IN OTHER DOCUMENTS Yes

18. RESEARCH LIBRARY THESAURUS

spoken dialogue systems, dialogue management, conversational agent, chatbot, human-computer interaction

19. ABSTRACT

A spoken dialogue system (SDS) is a specialised form of computer system that operates as an interface between users and the application, using spoken natural language as the primary means of communication. The motivation for spoken interaction with such systems is that it allows for a natural and efficient means of communication. It is for this reason that the use of an SDS has been considered as a means for furthering development of DST Group's Consensus project by providing an engaging spoken interface to high-level information fusion software. This document provides a general overview of the key issues surrounding the development of such interfaces.